Colour morphological sieves for scale-space image processing

Gimenez, David

Award date:
2007

Awarding institution:
University of Bath

COLOUR MORPHOLOGICAL SIEVES FOR SCALE-SPACE IMAGE PROCESSING

Submitted by
David Gimenez
for the degree of Doctor of Philosophy
of the University of Bath
May 2007

Signed 

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Abstract

Colour image processing, and to an extent multichannel image processing, have become increasingly important in recent times, largely due to the widespread availability of multichannel images. However, many of the methods used are still greyscale-based. This is to a large extent because of nonlinear filters, and in particular mathematical morphology, which are robust greyscale tools dependent on a explicit and relevant ordering which does not exist for data with multiple components.

This thesis considers the area of colour image processing using mathematical morphology. A brief review of the history and features of colour and image processing is given, with emphasis on the approaches towards image noise reduction and segmentation. Classic filtering methods are evaluated, showing their weakness and the room for colour morphology improvements; despite those problems, morphological greyscale scale-space sieves have found widespread application. Two recent colour morphology methods are then reviewed and proven superior, providing the motivation for the main area of work.

From these methods, new sieving structures have been derived from those and demonstrated to have similar or better denoising abilities than other morphological methods. The definition of classic attributes is also extended to colour; and a formal review of their segmentation performance is carried out, allowing a comparison against a set of well known edge-based segmentation techniques. The new sieves are a clear improvement over the greyscale methods, although they are outperformed by edge-based schemes. The combination of such methods with other techniques such as energy, vector attributes, the watershed algorithm and tree pre/postprocessing is also examined, showing their benefits and the potential for future work.

Contents

Nomenclature

HVS- Human Visual System
NMSE- Normalised Mean Square Error
MCRE- Mean Chromaticity Error
NCD- Normalised Color Difference
OSF-Order Statistic Filter
VMF- Vector Median Filter
TVMF/TNC- Truncated VMF
BVDF- Basic Vector Directional Filter
GVDF- Generalised VDF
AM- Area Morphology
AF/ASF- Alternating Filter/ Alternating sequential filter
SE- Structuring Element
σ^2- Gaussian variance
ρ- % of impulse corrupted pixels
L_x- metric distance for best neighbour selection
GSAOCS/GS-AOC- Greyscale extrema Area Open-Close Sieve
Y-AOC- Luminance extrema Area Open-Close Sieve
VAMS- Vector Area Morphology Sieve
CCS- Convex Colour Sieve
VAMM/VAMOCS- Vector Area Morphology M-sieve/ Open-Close Sieve
VDMS- Vector Distance Morphology Sieve
PCS M/V/A- Positional Colour Sieve, Marginal/Vector/Angular extrema
LCE/GCE- Local/Global Consistency Error
FoM(R,P)- Figure of Merit (Recall, Precision)
SSC- Stable Salient Contours

List of Figures

List of Tables

Chapter 1

Introduction

Image processing is the application of signal processing methods, using images as inputs and results. The first photographic image was produced in 1826 by Nicéphore Niépce after an exposure time of eight hours; latter processes (e.g. Daguerreotypes) gave an unique copy until the development photo negatives, produced in 1835 by William Talbot, that enabled multiple copies. Since this early point there has been a need for methods to improve the quality of images [1]. Up until a few decades ago, all processing was in the analogue domain and based on optics. This was mostly due to its inherent parallelism, the lack of affordable processing power (if any), and transmission means - the first "recording telegraph" or fax machine was not invented until 1843 by Alexander Bain and then implemented by Frederick Blakewell in the 1851 World exhibition, only becoming popular in the 20th century.

This changed in the 1960's with the need, and development later, of more advanced methods, primarily for satellite imagery and latter in medical, video, text, photography processing, at research centers such as Massachusetts Institute of Technology (MIT) and the Jet Propulsion Lab (JPL) at NASA among other [2]. More affordable computers in the 1970's, and the later availability of digital cameras to consumers in the 1990's, first analogue then CCD-based, and image compression further increased the importance of digital image processing. At the present most image-related activities involve some type of image processing, including satellite imagery, face recognition, medical imaging and meteorology to name a few. Still, analogue methods are still used in some niche applications as holography.

Greyscale image processing is now a relatively well established area of research, and techniques for segmentation and noise removal abound. In contrast, multidimensional image processing and in particular colour has only recently started to be fully explored, with most colour research being from the 1990's or latter as the required processing power became available. This is despite the first colour photo being taken in 1861 by the physicist James Clerk Maxwell and colour film being available from the 1930's. The most common approach of applying greyscale techniques to multiple channels fails to exploit the frequent correlation between different channels, can induce artifacts and often achieves less than ideal results, usually at higher CPU costs.

The advantages of colour are that more information is available, hence improving object discrimina-

tion , so it benefits automatic image recognition and analysis. In addition, the market for digital consumer cameras has greatly expanded in 15 years, reaching 22 million in 2003 and 96 million sales in 2005 [3], not including camera phones at 84 and 257 million respectively. This has motivated developments in colour-specific techniques. Mask based colour filters are well known (vector medians) and colour extensions of other greyscale methods (watersheds, snakes) are becoming standard tools for those involved in image processing. However, the extension of many other nonlinear methods to colour is an area of ongoing research.

1.1 Digital Image Representation

An image can be considered as a continuous intensity function $f(x, y)$ in terms of its spatial coordinates. Although it is possible to apply image processing at this point, the options available in that format, be it film or raster-scanned video, are rather limited (continuous transforms, equalisation, scaling and the like) and prone to corruption compared to their digital equivalents. Besides, digital methods include content analysis i.e. segmentation and feature extraction; hence the more usual step is to convert the image to digital form, giving an array of discrete values or pixels (picture elements). Doing so involves two processes, shown in figure 1.1:

- sampling, or taking data at regular intervals. This involves some lowpass pre-filtering to prevent aliasing by removing frequencies higher than half the sampling rate.
- quantisation, reducing the number and/or range of values a pixel can take, and hence the number of bits required to encode it.

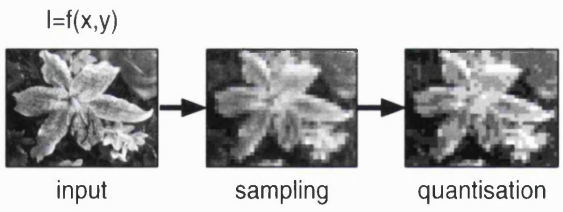


Figure 1.1: Image digitalisation

Figure 1.2: Colour image representation

The conversion from analogue to digital format can yield large amounts of data, so compression usually follows for storage and transmission purposes. The resulting image is a discrete array of pixels: in greyscale images, each pixel is a scalar quantity. An example is figure 1.3, giving bits/ sample versus downsampling ratio in each axis and final bit size. For multichannel images as figure 1.2, pixels correspond to a set of values, or a vector in n dimensions. Colour is a special case of the latter, with $n = 3$. Lately digital colour processing has gained more importance, partly due to the rise in digital camera devices, often requiring dedicated methods for colour processing.

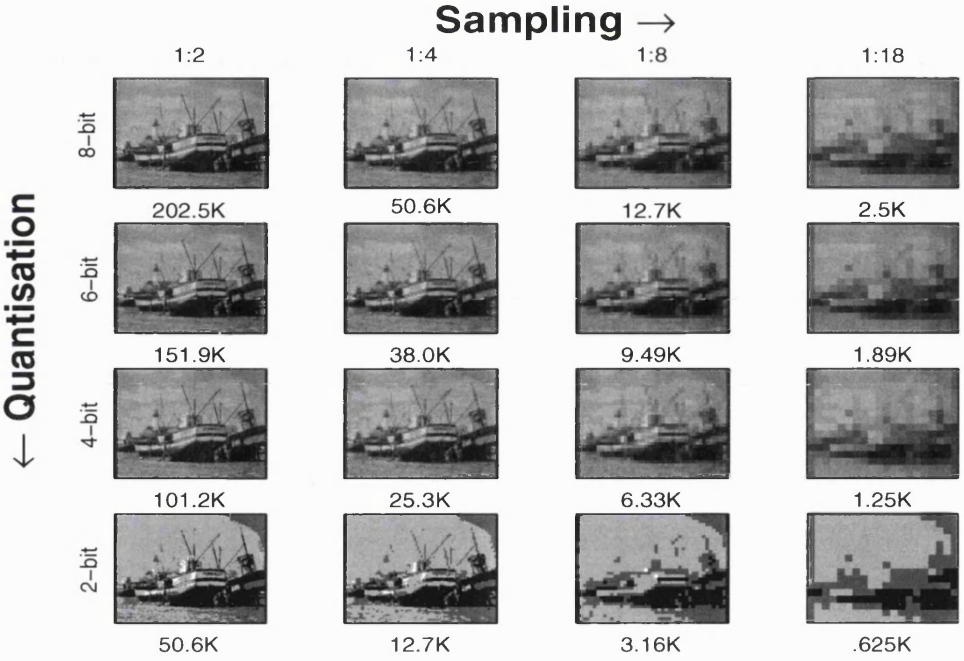


Figure 1.3: Sampling and quantisation effect, bit size for 'boats' (360*288, 8-bit greyscale, 810Kbits)

1.2 Colour Spaces

A colour space is the mapping of a range of colours in some standard way. The choice of colour space determines the properties measurable, hence the processing options, and the gamut or the range of values that are valid or visible. Most spaces have 3 components or degrees or freedom, related to the human visual system (HVS) having 3 types of detectors and perception being based on hue, value and saturation. 2-component systems are possible (e.g. RG, RGK, eigenvector decomposition), and similar sensitivity to red and green seems to hint so; however, chromatic distortion makes them impractical, as it can be proven by taking eigenvectors in image 1.5. The main ones are described below.

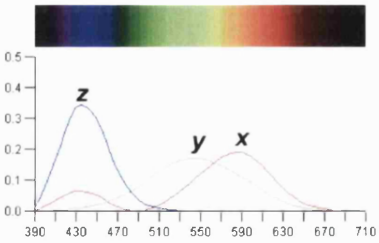


Figure 1.4: Human Colour sensitivity: receptor response with wavelength/ nm (graphics.lcs.mit.edu)

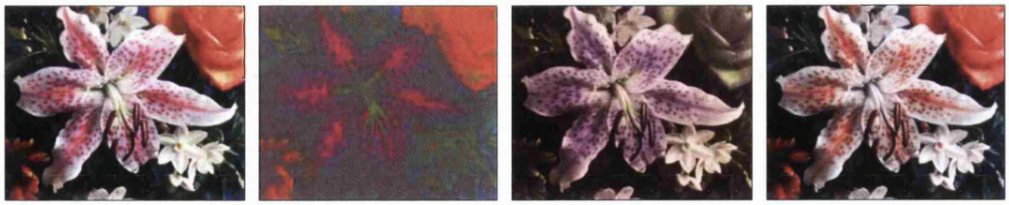


Figure 1.5: Effects of removing the 1st, 2nd or 3rd eigenvector (84%,14% and 2% of variance)

1.2.1 RGB spaces

The Red-Green-Blue (RGB) space is based on the physical detection and structure in the Human Visual System, or the strength of the colour component detected by each receptor (red, green, blue) in figure 1.4, all orthogonal and with equal ranges. Also other spaces can be linearly derived from it (YUV, YCC, YIQ, Ohta and others).

The main advantages are that all points in space are defined (no singularities), it corresponds with most physical equipment (in cameras as RGB, in monitor displays as RGB/YUV), removing the need for conversion, and the channel correlation means noise is spread among all channels, lessening its effect. It is often used in literature, as RGB [4] [5] or a transform.

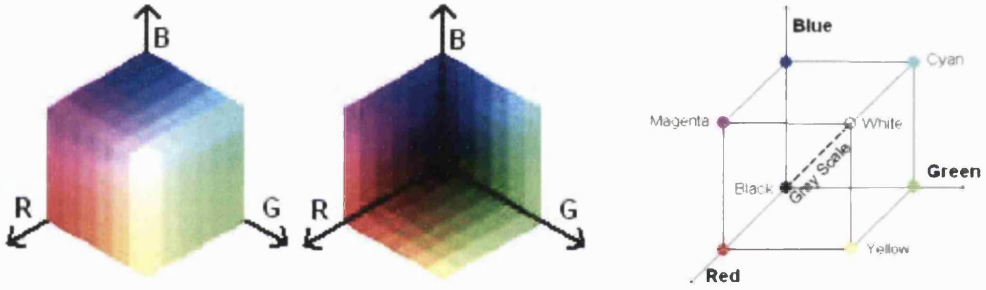


Figure 1.6: RGB model

Despite being a popular space, the RGB space has some shortcomings. There is high channel correlation, as luminance and most hues depends on their combined values, spreading information among all channels. However, this is an advantage against uncorrelated noise removal; most RGB-related spaces also bode well enough and those with a luminance component (YUV, YCC) solve this problem.

Another problem is that RGB triplets do not fit the subjective perception and intuitive definition of colours by hue/ luminance/ saturation. Also it is non-uniform, i.e. perceived distances are different from actual distances. This is partly redeemed in other derived spaces where luminance takes an independent component (e.g. YUV, Ohta), but the hue components needed still bring trouble.

Finally, the space is device dependent, different devices will have different sensitivity and tolerance, and without proper calibration the output is not universal.

1.2.2 HSI spaces

The family of HSI (hue, saturation, intensity) spaces is based on human perception, which treats colours as a combination of brightness, saturation and hue that are intuitive. Many spaces are available: HSI-HLS, HSV, HCI, HVC, TSD are a few, all named by their components. Common to all are the division in hue(H) or tone(T), the angular measure of the type of colour ranging from 0 to 360°; saturation(S) or chroma(C), the dominance of the hue component ranging from 0 to 1; and the brightness, luminance, intensity, value or even darkness component. The main variants are HSV and HLS.

The HSV space uses value, or the brightness of the largest component, giving a cone-shaped space with $V = max(R,G,B)$ and $S = max(R,G,B) - min(R,G,B)$

The HLS form uses intensity or overall luminance and an intensity dependent saturation, giving a bicone shaped space where L=50% gives the brightest gamut and L=100% gives white. This accounts for the fact white tones are perceived as brighter than saturated colours. The formulae are

$$\begin{aligned} L &= (max(R,G,B) + min(R,G,B))/2 \\ S &= \frac{max(R,G,B) - min(R,G,B)}{max(R,G,B) + min(R,G,B)}, L < 0.5 \\ &= \frac{max(R,G,B) - min(R,G,B)}{2.0 - max(R,G,B) - min(R,G,B)}, L > 0.5 \end{aligned} \tag{1.1}$$

Finally, the cylindrical HSV-HLS mapping is another variant rarely used; this normalises the saturation component yet makes the space less perceptually regular [6], having only some use in colour selection in computers.

In all cases, hue is defined as $H = atan((R - G/2 - B/2)/((G + B) * \sqrt{3}/2)$; another hue definition is available for the hex-cone variants of the HLS and HSV spaces which does not require the arctan function. These spaces are perceptually intuitive, however they have singularities: hue is undefined when saturation is zero, and it becomes sensitive to brightness changes at the extremes. In addition, the space is highly non-uniform, and conversion is complex. Finally, the hue component is continuous, which is a problem for extrema processing.

Despite these problems, it has been commonly used in literature [7] [8] [9] [10].

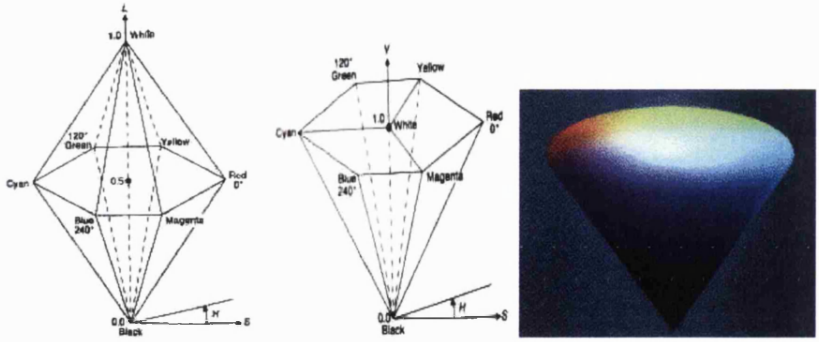


Figure 1.7: HLS and HSV models, and HSV representation (www.paris-pc-gis.com)

1.2.3 CIE XYZ

This space is derived from RGB and solves its device dependence. It uses 3 components, X,Y and Z, where Y is the luminance-oriented component while the other 2 carry the chrominance information.The main feature is that the primaries are chosen so only positive weights are needed to represent all visible colours, and therefore they lie outside of the visible range. These components can also be referred by x,y and Y, where x and y are the normalised chrominance and Y gives the luminance, giving the chromaticity diagram.

The XYZ space serves as a standard for equipment, what ensures the portability to similar devices. Different specifications are available based on the choice of the white point or illuminant [11]; some common ones are:

$$\underset{ccir601\text{-}1}{CIEXYZ}\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.606881 & 0.173505 & 0.200336 \\ 0.298912 & 0.586611 & 0.114478 \\ 0.000000 & 0.066097 & 1.116157 \end{pmatrix}\begin{pmatrix} Red \\ Green \\ Blue \end{pmatrix} \tag{1.2}$$

$$\underset{ccir709}{CIEXYZ}\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.412411 & 0.357585 & 0.180454 \\ 0.212649 & 0.715169 & 0.072182 \\ 0.019332 & 0.119195 & 0.950390 \end{pmatrix}\begin{pmatrix} Red \\ Green \\ Blue \end{pmatrix} \tag{1.3}$$

$$\underset{itu(D65)}{CIEXYZ}\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.430574 & 0.341550 & 0.178325 \\ 0.222015 & 0.706655 & 0.071330 \\ 0.020183 & 0.129553 & 0.939180 \end{pmatrix}\begin{pmatrix} Red \\ Green \\ Blue \end{pmatrix} \tag{1.4}$$

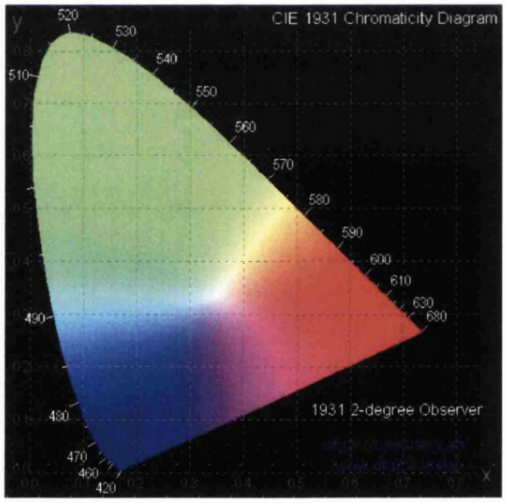


Figure 1.8: CIE 1931 chart for 2-deg observer (www.efg2.com)

This space is not directly used in operations, but rather is the base for others attempting perceptual uniformity.

1.2.4 LUV/LAB

Based on human perception models, they have similar formulas. They are derived from the XYZ space, given the parameters for the illuminant as a (X_o, Y_o, Z_o) triplet.

The spaces are quite, but not completely, uniform and can be used for subjective evaluations [12]. The conversion is complex, but can be reversed; they differ in shape.

Luv takes the form of a distorted cube with the greyscale diagonal forming the L coordinate, with the u, v components giving the chrominance and having different ranges. This is often used for colour addition (i.e. monitor displays).

$$L = 116 f(\frac{Y}{Y_o}) - 16 \tag{1.5}$$

$$\text{with } f(t) = t^{1/3}, \quad t > 0.00856 \tag{1.6}$$

$$= k\frac{903.3t^{1/1} + 16}{116} \quad \text{otherwise}$$

$$u = 13L(u\prime - u\prime_o) \qquad v = 13L(v\prime - v\prime_o) \tag{1.7}$$

$$\text{with } u\prime = \frac{4X}{X + 15Y + 3Z} \qquad v\prime = \frac{9Y}{X + 15Y + 3Z} \tag{1.8}$$

$$u\prime_o = \frac{4X_o}{X_o + 15Y_o + 3Z_o} \qquad v\prime_o = \frac{9Y_o}{X_o + 15Y_o + 3Z_o}$$

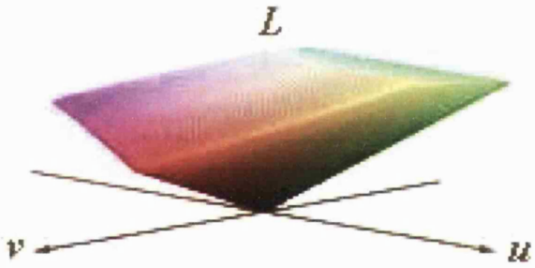


Figure 1.9: Luv model:L as the vertical axes,u and v clockwise

Lab takes a spherical shape, with the same luminance and the a,b components giving the red/green and blue/yellow chrominances. This is used for reflected (printed) light sources.

$$L = 116 f(\frac{Y}{Y_o}) - 16 \tag{1.9}$$

$$a = 500(f(\frac{X}{X_o}) - f(\frac{Y}{Y_o})) \tag{1.10}$$

$$b = 200(f(\frac{Y}{Y_o}) - f(\frac{Z}{Z_o})) \tag{1.11}$$

$f(t)$is as for Luv

Both spaces are perceptually uniform (Euclidean distances relate to perceived changes over short distances) and have been used often in literature [13].

1.2.5 Choice of space

This project concentrates on the use of RGB. The main reason is most input equipment and data storage deals with the RGB space, or a space easily derived from it; hence no transforms are required beforehand, and on an application point of view this is best. Also as most colour images are captured via RGB devices, it allows noise to be accurately modelled as partly correlated, using the same space as practical devices for noise modelling seems reasonable.

Another reason is the RGB space contains no singularities, unlike the hue for achromatic vectors in the HSI family of spaces; in addition, the RGB components have equal importance. This allows enough flexibility to apply any methods found to N-D spaces (multi-frequency SAR, velocity fields) without changes.

Finally, some of the measures used consider the angle between vectors, taking the space origin as the vertex, for hue-saturation. This is limited to $\frac{\pi}{2}$ rad in RGB, but increases for luminance-oriented spaces with bipolar components, and proves less adequate for Lab/ Luv spaces; in the latter cases, the individual chroma components are a better measure of hue.

1.3 Noise types and noise metrics

Noise is unwanted information in the image, and the inclusion of such information, whereas distortion refers to changes due to processing such as filtering or channel frequency response; both reduce the image quality. Noise sources include ADC operations, amplification, sensor noise (often removable by increased or averaged expositions) and transmission losses, and can be classified as additive (overlaps the input) or multiplicative (modulates the scaling factor at some point). The 3 main types of additive noise considered in this project are

- white Gaussian, zero mean: $f(x) = e^{-\frac{x^2}{2\sigma^2}}$
- impulsive or salt & pepper noise: $f(x) = \frac{1}{2}\delta\{|x| - 1\}$
- a mixture of both Gaussian and impulsive

Those were chosen as they represent well enough most noise phenomena: Gaussian noise represents well the thermal noise at most analogue stages and the photon noise in a CCD, whereas impulsive is a good fit for transmission faults and CCD readout errors. Note that other distributions exist, like Poisson, uniform or coloured Gaussian, but at moderate levels their effects are all similar to Gaussian. These signals are usually partly correlated due to their natural origins, since:

- in RGB systems, the most common type for input devices, the luminance information is distributed in all 3 channels, and hues other than the primaries need different channels, hence information is spread between all channels, both wanted and unwanted (scatter, reflection, photon noise at short exposures or low lighting). Other noise sources in CCD, like dark noise (dependent on device temperature) and read noise (due to ADC of sensor information) [14] are independent of the input.

- most RGB input devices have an array of mixed CCD sensors for each band, rather than 3 uniform sensor arrays; this requires interpolation before obtaining the output, thus multiple channels can be affected by the same noise source and colour artifacts often appear in other cases. Also overlapping bands of sensibility for each sensor can spread both information and noise among channels and local effects like blooming (saturation) [15] are more likely to affect adjacent sensors than farther ones.
- transmission noise after quantisation can affect adjacent samples when 'bursty' channels are considered; the effect depends on prior data encoding.

There is also an element of uncorrelated independent noise in each channel, due to amplification/ post-processing in each channel. At most points, a simple uncorrelated model with equal noise probabilities in each channel is used. This more common model means a higher number of pixels than stated are affected. For impulsive, this means $1-(1-p)^3 \approx 3p\%$ colour pixels affected for $p\%$ of corrupted points in the image, with the same distortion metrics as for fully correlated noise. That gives 2.97% for 1% data affected, 14.3% for 5%, 27.1% for 10%, etc. For Gaussian, this only increases chromatic distortion slightly.

A more complex model [16] is used at points, giving a mixture of uncorrelated (1 channel only) and correlated (all channels) noise at probabilities ε and $1-\varepsilon$. Taking N as noise in one channel, impulsive noise is defined as

$$I'_{x,y} = \begin{cases} \{I_r(x,y), I_g(x,y), I_b(x,y)\}, & 1-\rho \\ \{N, I_g(x,y), I_b(x,y)\}, & \rho * \varepsilon/3 \\ \{I_r(x,y), N, I_b(x,y)\}, & \rho * \varepsilon/3 \\ \{I_r(x,y), I_g x, y, N\}, & \rho * \varepsilon/3 \\ \{N, N, N\}, & \rho * (1-\varepsilon) \end{cases} \tag{1.12}$$

Gaussian correlated can be introduced by the covariance matrix in equation 1.13, normalised by the variances of each given channel pair, i.e. $\overline{\delta^2_{rg}} = \frac{\delta^2_{rg}}{\delta_r \delta_g}$. All instances of uncorrelated Gaussian are independent on each channel, giving $\varepsilon \simeq 1$, and $\varepsilon = 0$ corresponds to identical noise signals.

$$\overline{\delta^2_{rgb}}(N) = \begin{Bmatrix} 1 & 1-\varepsilon & 1-\varepsilon \\ 1-\varepsilon & 1 & 1-\varepsilon \\ 1-\varepsilon & 1-\varepsilon & 1 \end{Bmatrix} \tag{1.13}$$

1.3.1 Image quality assessment

Any type of result, be it the output of a filter or the corrupted version of an image, needs to be measured and compared to others in order to evaluate the effect of filtering. Here are some of the metrics considered and employed.

1.3.1.1 Qualitative measures

Mean Opinion Score is a popular subjective metric, like Receiver Operating Curves. The main inconvenience of these measures is collecting enough feedback to make them representative. Therefore such

measures were not used for noise (where those related are not likely to show any difference) but for segmentation goodness, as part of the Berkeley database [17] [18].

1.3.1.2 Quantitative measures

Metrics considered in the project are the NMSE, MCRE and NCD [19]. Others, such as RMSE or MSE, have been rejected as they are more appropriate for greyscale images.

- NMSE: The normalized mean square error, the MSE of every image component divided by the MSE of the original image. This is fairly sensitive to brightness and chrominance changes, response to the latter one depending on the actual luminance.

$$NMSE(f,g) = \frac{\sum_{x=1}^{M}\sum_{y=1}^{N}(f_{xy}-g_{xy})^2}{\Sigma_x\Sigma_y {f_{xy}}^2} \tag{1.14}$$

- MCRE: The mean chromaticity error is the square root of the MSD of the normalized vector images, this is more sensitive to hue and saturation changes. Also it considers the distance in the Maxwell triangle between pixels- this is the variant used, although others [20] [21] have been devised. In all cases, this proves independent of magnitude changes, so other metrics are required to describe them.

$$MCRE(f,g) = \frac{\sum_{x=1}^{M}\sum_{y=1}^{N}\sqrt{(r_{xy}-\overline{r_{xy}})^2+(g_{xy}-\overline{g_{xy}})^2+(b_{xy}-\overline{b_{xy}})^2}}{MN} \tag{1.15}$$

- NCD: The normalised colour difference is based on the use of the LAB/LUV spaces, which depends on the ITU recommendation being used [12]. This metric is sensitive to both luminance and chroma changes and closely reflects small perceptual differences, yet the presence of multiple standards proves a problem. In any cases shown, the D65 white point has been used (see equation 1.4).

$$NCD(f,g) = \frac{\sum_{x=1}^{M}\sum_{y=1}^{N}(||f_{xy}-g_{xy}||_2)}{\sum_i\sum_j {f_{ij}}^2} \qquad ||f_{xy}||_2 = \sqrt{L_{xy}^2+u_{xy}^2+v_{xy}^2} \tag{1.16}$$

1.4 Colour Segmentation methods

Segmentation is the process of converting a set of pixels into meaningful clusters corresponding to objects. As mentioned by [22] these can be split into:

- feature-based or grouping according to their overall division in groups of distinct properties: includes clustering, histogram thresholding.
- image-domain or partition according to their spatial connection and coherence: includes split-and-merge methods, region growing (by merging or by expanding from initial seeds), edge detection (including snakes or energy-minimizing snakes) and neural networks.

- physics-based or a variation of the above, based on reflection and light distortion by objects and exclusive to colour (more domain-specific, like [23], which uses a human vision model for thresholding).

Region-based methods provide closed boundaries and regions with connected pixels, edge-based give relevant boundaries which must be connected afterwards, and cluster-based find histogram clusters giving objects by their vector value. Some work has been on combined approaches (eg [24]), but in this research only region-based methods are considered. See [17] [25] for edge-related evaluations, [22] [26] [27] for region-related ones.

1.4.1 Edge-based

Edge-based methods take boundaries or gradient fields; the natural correspondence between object boundaries and gradient maxima gives relevant boundaries when thresholded. Early examples include Roberts, Sobel, LoG and Canny edge detectors, all based on fixed-size masks. More recent ones, like the generalised compass operator [28] or combined approaches [17] [29], are not limited to one scale. Ruzon's operator [28], although related, is not the earlier compass edge detector [30] based on fixed kernels, such as Sobel masks. In most methods the output is often broken in segments (by thresholding or lack of an actual edge), which must be linked to give closed boundaries; Leung and Malik [31] expose subjective edges (as no gradient boundaries) as a means to fill the gaps, giving relevant regions.

Another option is active contours i.e. energy snakes [32] [33]; these are an iterative method, affected by local minima and initial conditions, although proper routines can deal with this problem. The general formula for a contour, or spline, is: given an image $G(x, y)$ and a contour (not necessarily closed) on it with r points, the energy of the contour is

$$E(r) = \int_r (\alpha|C'(q)|^2 + \beta|C''(q)|^2)dq - \int (\gamma \nabla G(C))dq + E_{ext} \tag{1.17}$$

The 3 main energy components at any point are:

- the internal component, giving a minimum at the most regular and compact shape. This has two components: $C'(r)$ or the distance between points, and $C''(r)$ or the degree of curvature.
- the gradient component, opposed to the other two and being optimal at the points of highest gradient, $G'(x, y)$
- the external component, due to any constraints. This allows attracting or repelling the spline from any given point, restricting it within a region of interest or any other configuration. For example, $E_{ext} = -\sum_r k_i|p_1 i - p_2 i|^2$ [34] lowers the energy of p_2 in the snake when near p_1 on the image.

Taking all the weighted components over the entire contour gives the local energy; differentiation of the formula gives the base for an iterative method. Another option given by [35] is the water-snakes, a multiscale, watershed-based energy snakes method. The watershed, although based on a gradient, is a region-based method; however, many edge and region-based modifications have been developed and more generally, edge processing is often watershed pre-processing.

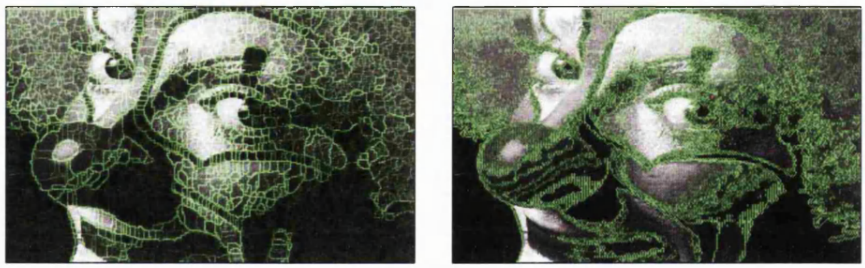


Figure 1.10: Watershed, morphology segmentations

Generally the main problem in these methods is the lack of closed boundaries and the dependence on a clear step between two adjacent areas. For gradient methods see [33] [36] [37] or [4] [38].

1.4.2 Region-based

In this category fall tree-related methods, multi-scale methods (split-merge, quad-trees, pyramids and others capable of scale-spaces) [39] and region growing; although related, diffusion-based scale-spaces need an edge detection-linking step. In all cases, connected pixels with some similarity are grouped as one region, giving spatial coherency. Region-growing methods take a number of seeds and grow the regions to segmentation (e.g. the watershed, or mathematical morphology), whereas tree/ graph methods take large numbers of regions to merge after an initial splitting stage, allowing for multi-scale representations after successive cycles of downsampling and smoothing (except for quad-trees, which only subsample). In tree/graph methods other than split-merge ones (e.g. quad-trees with fixed splitting) the initial pixels are often the starting point barring identical, connected neighbours; these data structures and morphological sieves allow for scale-spaces, with a continuous scale parameter rather than in multiples, giving a better concept of the image as such.

A popular method is mathematical morphology sieves; until more recent years this was not too popular in colour, beyond different channel orderings and marginal processing. These methods prove somewhat robust to partial occlusions [40]. Region growing until reaching inhomogeneity has been another popular constraint besides area [41], in particular in split and merge methods. More detail on morphology follows in latter chapters.

The watershed is another region-based morphological method; regions are determined by the minima in a gradient field. It is also one of the most popular methods since it gives closed boundaries for all its regions: its main problem is over-segmentation, on which a great deal of work has taken place, whether in pre-processing gradients or merging watershed basins by their differences, giving the recursive waterfall algorithm [42]. Also many works take the watershed basins as initial regions, followed by standard region merging. Colour extensions of this method often involve using multiple gradients or a gradient on vector distances. For more information on watershed and watershed-based methods (with post-processing), see [4] [8] [10] [35] [43] - [51].

A more recent development is normalized cuts [52]. This method considers the data as a connected graph or net of points, akin to mathematical morphology, but the distances are a function of the features

and the aim is to minimize edge weight among the subgraphs relative to their total edge weight. This is a more stable but expensive measure than min-cuts, giving good results [53] for dissimilar features [54].

One common weakness of this type of methods is that, unlike clustering, only connected regions can be grouped together, failing to identify regular patterns. For tree/morphology techniques, see [55] - [61] and [31] [62] - [67]. For other region-based and video processing methods see [68] -[70] , and [71] -[75] for mixed methods.

1.4.3 Clustering-based

The general concept of clustering is to group data by their features (be it data value or local patterns), pick representative groups with minimum overlap and assign them values accordingly. Often a regional coherency step is introduced to avoid small variations, and this has been extended into textures in colour [76]. Classic examples include fuzzy means, n-mean clustering and histogram thresholding.

Some of the difficulties with these methods arise from the distribution models used (especially if multiple groups overlap each other or their number is undefined) and the lack of spatial coherence. For example, histogram clustering fails for unimodal distributions (as in a set of objects without step changes in intensity/ distribution). Kamvar et al. [77] gives a thorough discussion on such limits.

However, like Frederix et al. [78], there are now non-parametric clustering methods, giving better performance; their method is also different by including spatial information despite allowing for n-dimensional clusters. Also the recent use of region and especially graph-based methods such as min-cuts [53] solves many of these problems, handling sparse data with ease. For histogram/fuzzy clustering in colour, see [5] [9] [13] [79] [80] [81] [82]; for texture classification [83] [84] are recommended.

1.4.4 Neural networks

Neural networks (NN) are a multi-layer structure of nodes or 'neurons', whose output is dependent on the weighed inputs received from nodes in previous layers according to the firing rule [85] [86]. These can be trained to detect clusters, or their combination, giving more options for finding groups of data than basic clustering. Although the most common use of NNs is extraction of more complex patterns or groups than general clustering, see [87] or [88] for examples beyond clustering, such as edge detection and region segmentation

The most common type of network is the perceptron, usually being a feed-forward network (no feedback loops present), which must undergo a training stage during which the weighting is modified to give optimal output as compared to the optimal one; the best known supervised learning method to do so is the back propagation algorithm, excelling at discrete classification. Another network type, the Kohonen self-organising network, is a self organising feature map (SOFM) and an example of unsupervised learning. Its aim is finding natural patterns rather than comparing to the training data, mapping the data structure onto the network map, i.e. similar input vectors affect nearby/connected states.

The main advantages of this technique are the self-organisation, flexibility and reliability it can achieve after training. This, in turn, is the main disadvantage, since the training set, number of iterations and network size must be large enough to extract significant features and give a consistent output.

At the same time, too large a network can cause overspecialisation, i.e. responding only to the training data [89].

1.5 Summary

Colour has been an active area of research since the 1990's and especially in the last years. There has been much work at areas such as colour models and the extension of greyscale techniques with some success. However, there is still room for improvement in the areas of region-based segmentation and nonlinear filtering, whose performances degrade when applied to multiple channels; this thesis explores that area.

The properties of nonlinear and morphological filters and the problems associated with colour are outlined in chapters 2 and 3. Chapters 4 to 6 presents a new colour alternative to mathematical morphology, exposing the options available beyond greyscale morphology and defining two new methods. Finally, the segmentation goodness of all the methods is evaluated in chapter 7, with overall conclusions in chapter 8.

Chapter 2

Nonlinear filtering techniques

2.1 Local Window Processing filters

Many basic image processing methods are based on masking, or taking the population within the local window at the point of interest, often a square mask. These then filter or take a new, maybe better, estimate of a pixel from the adjacent data, based on the spatial correlation between adjacent samples. There are two main filter classes, linear and non-linear. Linear methods obey the principles of homogeneity and superposition, in other words $f(ax) = af(x)$ and $f(x + y) = f(x) + f(y)$, and are often derived from polynomial formulae, frequency transforms or FIR filters; often they can be decomposed into multiple filters (e.g. 2-D Gaussian mean as two 1-D Gaussian filters or cascaded lowpass filters). They suffer from a smoothing effect and noise blurring.

Non-linear processing and especially order-statistic filters [90] [91] have become more popular methods of noise removal than linear methods, namely by their ability to remove noise with reduced edge blurring, as:

- linear methods suffer from edge blurring, and in some cases ringing, due to finite slope rates; also new vector values are produced which might not be valid in the space.
- non-linear methods allow more variety: rank-order filtering (morphology, median filters), energy minimisation, adaptive averaging, rational filters (non-linear polynomials or a ratio of polynomials [92]), etc.
- unlike rank-order filters, linear methods are weak against impulsive/non-uniform noise, that is unless the noise spectrum does not overlap that of the original image.
- the HVS has a nonlinear response, roughly logarithmic with contrast and frequency-dependent. Nonlinear filters can then offer better noise removal or remove undetectable information, thus enhancing compression.

For those reasons linear methods are not further examined in this thesis. The more basic nonlinear methods are the median filters: types are normal, weighted, truncated, adaptive or trimmed to name a few

[93] [94] [20]. Linear methods can be extended to colour without further work channel-wise; rank-order filters cannot, and are examined later in this chapter.

2.2 Order-statistic filtering

The review of prior methods was first focused on window-based processing techniques, especially non-linear methods due to all the reasons discussed previously, essentially lesser edge blurring, more variety and better noise response. The main types are rational and order-statistic filters (OSF); these include morphological, stack, polynomial, homomorphic and rational-hybrid among others.

Polynomial filters are defined by a polynomial of any points in the local window, extending the formula of basic (lowpass) FIR filters to 2-D; the more general rational filters [92] follow the same approach with multiple polynomials. Neither was considered; potentially giving better results in problems like interpolation and edge preservation [95], they also are more complex and less robust, therefore only order-statistic were studied. The general equation for order statistic filters is

$$OSF = \sum_{i=1}^{N-1} \alpha_i x_i, x_i \leq x_{i+1} Gain = \sum_{i=1}^{N} \alpha_i \tag{2.1}$$

These filters encompass methods like medians, skewed medians, trimmed means and mathematical morphology to begin with; the difference is in the coefficients, which for the medians and mathematical morphology are binary (only one value is nonzero). For the median it is the middle one ($x_{(N+1)/2}$), and in morphological erosion it is the first (x_0).

The OSF methods studied were vector medians, multiple distance metrics for vector medians and, later on, extensions of structure-based mathematical morphology (MM) to colour. These have become popular ways to deal with colour, if not ideal for MM, and both require vector ordering.

2.2.1 Median filtering

Median filters are order-statistic filters [96]. This type of non-linear filter is a good alternative to averaging filters when blurring is to be avoided or no new values can be introduced in the output. It is also the best filter against biexponential noise, and is robust against impulsive noise: this is due to the step-edge function being one of its root signals. The median filter takes its output as the value that minimizes the total distance to all the neighbours

$$d_{Xmed} <= min(d_n), n \in N \qquad d_n = \sum_{j}^{N} |w_j| * ||x_n - x_j||_a \tag{2.2}$$

which for scalar inputs is equivalent to taking the middle ordered sample

$$X_{med} = x_{(n+1)/2}, n \in N \qquad x_0 \leq x_1 \ldots \leq x_N \tag{2.3}$$

In both cases the neighbourhood is defined by a mask of area N, and by making equation 2.2 an

equality or avoiding even N values in 2.3, no new values are generated. In the case of the standard median, all weights in the first equation are equal.

For the weighted and adaptive medians, weights give different importance to each value; integer weights correspond to multiple instances of the input in greyscale, which are written as $W_j \diamond x_j = \{\underbrace{x_j, x_j \ldots}_{W_j}\}$.

Example: given the data {0,2,10, 0,5,4, 1,6,6} and taking the median,

- ordering gives {0,0,1,2,4,5,6,6,10}. The normal 1-D median is the 5^{th} value, or 4.
- Using weights of {1,1,1, 1,3,1, 1,1,1} gives a weighted median data of {0,2,10 ,0,5,5,5,4, 1,6,6}. That gives a median of 5 instead.

Some of the problems associated with the median are the weakness to Gaussian noise and other effects that effectively alter most pixels, not just some of them, the need to iterate for idempotent or invariant results and other problems noted by Davies [97] [98], including the rounding and shifting of edges at higher scales depending on the mask shape and area (the median will eventually remove local features less than 1/2 the mask size) and the weakness to noise patterns which are a root signal, e.g. a stream of impulses in the 1-D median.

2.3 Data ordering for Colour Nonlinear processing

Many nonlinear methods depend on the use of probability statistics (such as variance in adaptive filters) or the use of rank statistics (as maxima in openings and the median in median filters), unlike in linear methods that are shaped as polynomials. In the former, ordering the data is required, which is not obvious in colour spaces. Ordering approaches for multidimensional data are divided as follows:

- marginal: taking each channel as independent greyscale images and processing as required. This can lead to edge blurring and creation of new colours.
- conditional: taking a metric derived from one of the input channels, or a combination, as the ordering parameter. Also known as lexicographic order, this is a one-to-one transform, allowing for full ordering; the first metric tends to dominate the ordering. This can be weighted and truncated [99], changing the bias from the first variable.
- reduced: taking a combined metric of the input vector as the extremeness. This is a distance metric, and a many-to-one transform; no distinction is made between different types of outliers. These comprise distance ordering, taking the overall metric to a set of neighbours, or projection ordering, with constant conversions for each input.
- partial: take the spatial outliers of a set by fitting a hull/cluster, then repeat with the remaining points. This creates a set of concentric groups of gradually decreasing outlierness, like in reduced ordering. However, ordering within each set is not defined, and hulls are expensive. This is related to global clustering methods.

Example: Let the input vectors be {(0,0) (1,2) (3,1) (5,2) (7,6)} and the median is required. Note the median is the $\frac{n+1}{2}$ point out of n ordered points, or that with the least total distance to all others:

- marginal ordering gives the ordering {(0,0) (1,1) (3,2) (5,2) (7,6)} and a median of (3,2), an entirely new value.
- conditional ordering on the second channel, then the first gives the ordering {(0,0) (3,1) (1,2) (5,2) (7,6)} and median (1,2).
- reduced distance ordering on the city distance metric ($L_1(x,y) = \sum_i = 1^D |x_i - y_i|$) to all points gives distances of (27, 20, 19, 20, 38), or an ordering of {(3,1) (1,2) (5,2) (0,0) (7,6)}. The median is (3,1), since the aggregate distance is the lowest.
- partial ordering by removing the point furthest from the mean of the unselected remaining points removes (7,6), then (5,2), later (0,0) and (3,1), and last (1,2). The ordering is {(1,2) (0,0) (3,1) (5,2) (7,6)}, and like in reduced distance ordering the median is the innermost point (1,2).

The mean of the distribution is (3.2, 2.2), and the median is generally a close value; only the marginal and reduced ordering select close values. Note the dependence of conditional ordering on the rules used: using the channels in opposite order gives a median of (3,1). This shows the other problem of vector ordering: the best metric depends on the application and vector space used.

Hence in colour spaces taking the median sample is not possible: marginal ordering induces new vectors and edge jitter, and conditional ordering gives hue artifacts. For morphological filters it is even worse, as vectors can be maxima and minima in different channels, be neither when conditionally ordered and yet be clearly uncommon.

That leaves definition 2.2 with reduced ordering [16] [100] [101] as the best option. Some work has been done in median extension to higher dimensions and colour images, so a set of known vector filters [19] [20] [102] [103] was studied. Other types of medians, like weighted medians [104] or hybrid mean-medians were excluded.

2.4 Vector Median filtering

2.4.1 Vector Median Filters selected

As stated in the previous subsection, there is a large number of vector median filters. For simplicity, only metric variants of the basic vector median were considered, excluding all weighted and adaptive weighted median filters (AWM). These are more dependent on mask shape and weighting, and image contents, than other filters, giving too many options to consider; in addition, both options can be applied to any variation of the vector median considered here.

The filters described are the basic vector median filter (VMF), along with some of its variants. All these filters were tried with a square mask due to computational efforts with circular ones, time invested and the reason the comparison was centred on their inherent behaviour, not their possible optimisation.

2.4.1.1 VMF

The classic VMF takes as its output the point which minimizes the sum of distances to all other points in the window (see equation 2.2 and figure 2.1). This was studied with city distance(L_1), Euclidean(L_2), square Euclidean(L_2^2) and luminance(L_3) metrics; the latter one proves of little use, other than in the GVDF calculations.

Also included are fast versions for the L_1 and L_2^2 metrics, using a reduced metric [105] [20]: the metric to be minimised is the distance to a centroid similar to the actual median (the marginal median or mean, respectively), not the overall distance to all points. This provides a visible speed gain in calculations with little difference to the full median; taking figure 2.2 and the square Euclidean, the VMF with total distance gives weights of $(32, 36, 33, 66, 89)$ and an ordering (A, C, B, D, E), identical to that taking the centroid distance to the mean (2.2,2.2). The general formulae are

$$VMF = min(\sum_{i=1}^{N} L_x(x_i, x_j)), i, j \in (1 \ldots N) \qquad \text{with metric } L_x, \text{ or} \tag{2.4}$$

$$VMF = min(L_x(x_c, x_j)), j \in (1 \ldots N) \qquad \text{with metric } L_x \text{ and } x_c \text{ centroid} \tag{2.5}$$

$$L_x(a, b) = ||\vec{a} - \vec{b}||_x$$

for the full and reduced medians respectively, where L_x stands for the distance (Euclidean, angular, or others) and N is the pixels selected by the mask. Chatzis et al. [106] give the fuzzy vector median filter, which is somewhat related to this vector median. The main advantage of the VMF over greyscale is the lack of artifacts and other effects; this method is also more intuitive when using a weighted or adaptive mask.

The other filters described are the basic and generalized vector directional filter (BVDF, GVDF), a variant of the directional distance filter (DDF), the truncated vector median (TVMF), and a basic version of the adaptive weighted median (AWM).

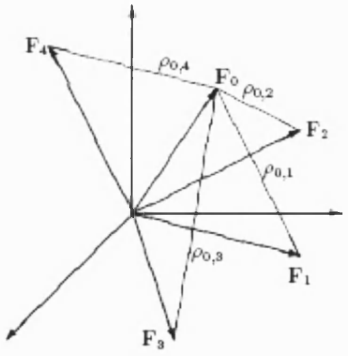


Figure 2.1: VMF illustration (cnap.polsl.pl/DIP/labs)

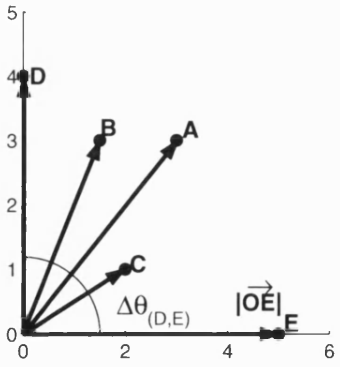


Figure 2.2: GVDF illustration

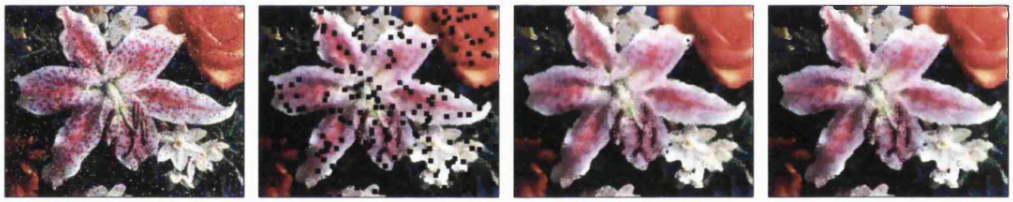
2.4.1.2 GVDF

This filter is derived from the VDF/ BVDF [102]. The method begins with the median using the vector angle metric, followed by luminance selection of the first M vectors with the least angle. Therefore,

$$BVDF = min(\sum_{i=1}^{N} L_\theta(x_i, x_j)), \qquad i, j \in (1 \ldots N) \qquad L_\theta(x_i, y_j) = \arccos(\tfrac{x_i . y_j}{|x_i||y_j|})$$
$$GVDF = d_{(M+1)/2}, \quad d_r = L_2(x_r, x_j), \quad d_0 \leq d_1 \leq d_2 \ldots d_{M-1} \qquad , M <= N \tag{2.6}$$

Taking figure 2.2, the BVDF ordering of vectors is (A, B, C, D, E) whereas their magnitude ordering is (C, B, D, A, E). The BVDF will select A as the median, while the GVDF will first take A, then B and then D when selecting from 1,3 or all values.

L_2 magnitude selection was used instead of the L_1 magnitude, since this measure models nearer to the HLS space and gives a slight improvement in metrics and quality. This filter is best where luminance is not low or greatly varying, since hue differences are less representative at those points; in the limit of M=N, this filter becomes a greyscale median with luminance conditional ordering, whereas for M=1 the GVDF becomes the BVDF. As an extra note, the L_θ metric implemented considers the 'black point' or origin to have random hue, and hence to have the largest possible angle to any other vectors, see figure 2.3d. This, although biased against that colour, prevents the artifacts of dark areas appearing around all original black spots (if considering the origin not to have an angle difference, fig.2.3b) or just those near white/grey regions (when assigning it a neutral colour, fig.2.3c) and improves the NMSE against impulse or mixed noise by 10%.



(a) input (b) any (c) neutral (d) none

Figure 2.3: BVDF filtering of 'lily' with 10% partly correlated noise (a), and hue for origin

2.4.1.3 DDF

The standard DDF is defined by taking the product of the sum of the distance and angle metrics and minimizing it [104]. This is more effective than the initial approach taken here, minimising the weighted sum of the metrics rather than the product.

Later, a weighted DDF formula was used as described in equation 2.8. Results still proved more sensible to noise than in other approaches. This was due to the sensibility of the angle to low luminance regions, affecting luminance edges where angle is sensible to noise. From actual results and noise

performance, the best ratio is a 20-50% of BVDF to VMF.

$$DDF = \min(\sum_{i=1}^{N} L_\theta(x_i, x_j) \sum_{i=1}^{N} L_2(x_i, x_j)), \qquad i, j \in (1 \ldots N) \tag{2.7}$$

$$DDF(\rho) = \min\{(1 - |2\rho - 1|) * DDF(x_i, x_j) + \ldots \tag{2.8}$$

$$\begin{cases} (2\rho - 1) * \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{N} L_2(x_i, x_k)) * \sum_{i=1}^{N} L_\theta(x_i, x_j) & \rho \in (0.5 \ldots 1) \\ (1 - 2\rho) * \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{N} L_\theta(x_i, x_k) * \sum_{i=1}^{N} L_2(x_i, x_j)\} & \rho \in (0 \ldots 0.5) \end{cases}$$

2.4.1.4 TVMF

This filter attempts mode filtering by approximation. Since finding the mode, or most common value, of a population from the histogram has proven elusive, this takes the idea of approaching the mode by using the mean and median as references like in the truncated median [94], then extending it to higher dimensions [20] as in figure 2.5. In scalar data with smooth distributions and one main peak as in figure 2.4, the median always lies between the mode and mean; trimming only the values on the mean's side removes those outliers, shifting the median closer to the mode. With vector data the mean-median axes is less reliable, but the same approach to remove the most extreme edge of the distribution is used: taking the median (M_1 in figure 2.5) and the main outlier N by reduced ordering gives an axis with P, opposite to N and equal distance from M_1 (2.5b). The outlier T in the opposite direction is the closest to P (2.5c). Cropping the points farther from M_1 than $\overline{M_1 T}$ and getting a new median M_2 (2.5a,d) is trivial.

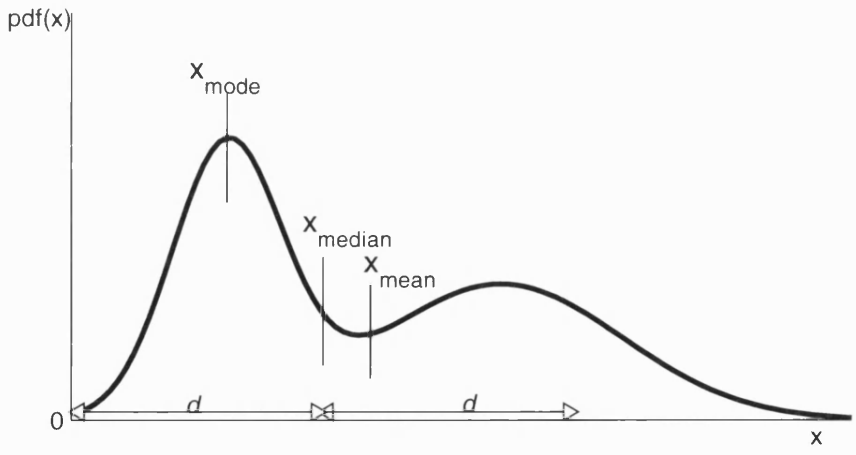


Figure 2.4: Mode finding by truncating from the median

This sharpens the results of other median filters, giving better edges and noise removal at any scale. The costs are worse objective metrics and higher calculation times, since it works recursively to find the mode and the boundaries get gradually sharper, which is a good point other than for those metrics. In theory the filter must iterate until full convergence; in practice, as stated by Davies, only one iteration is required, since only minor, visible changes appear up to 3 iterations; for the Euclidean square metric, two iterations are required before no further changes are perceived, possibly due to the blurring induced.

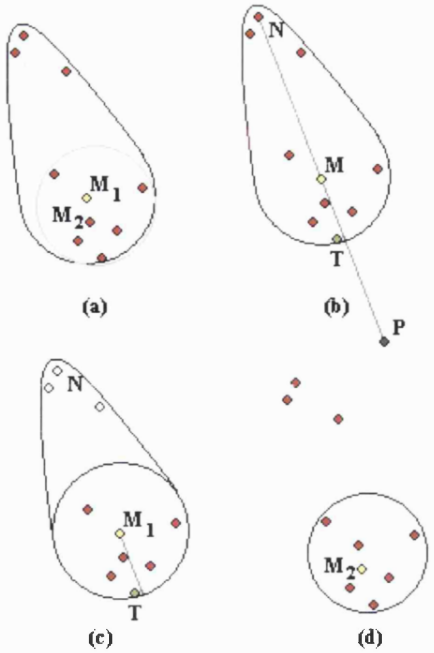
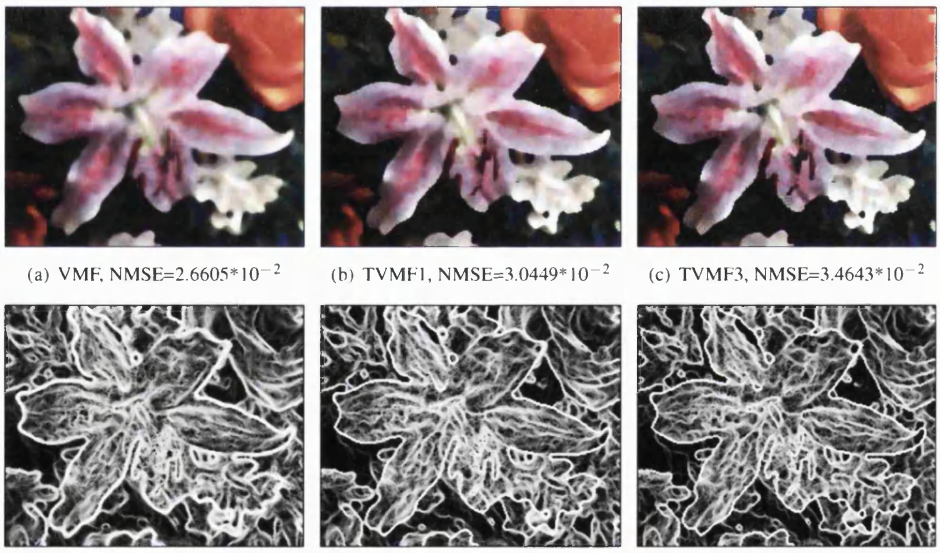


Figure 2.5: Truncated median construction: M_1= original median, M_2= truncated median



(a) VMF, NMSE=2.6605*10^{-2} (b) TVMF1, NMSE=3.0449*10^{-2} (c) TVMF3, NMSE=3.4643*10^{-2}

(d) VMF gradient (e) TVMF1 gradient (f) TVMF3 gradient

Figure 2.6: Truncated median; effect on image and equalised gradient fields

This filter was developed based on the reduced VMF, therefore the choice of metrics was limited to the L_1 and L_2^2, but was enough to see its potential with other metrics. In figure 2.6 above, the 1 iteration

7*7 TVMF has 10061 points above the mean with an average edge image value of 112.4, and the 3 iteration 7*7 TVMF has 9873 points above the mean with an average value of 127.0, compared to 12068 points with an average of 81.5 in the 7*7 VMF, clearly enhancing edges.

2.4.1.5 AWM

All of the filters above suffer from not taking into account image contents, and the need for lesser smoothing near edges or fine detail; in addition, all vectors are considered equally when points near the window center are more relevant. The adaptive weighted median (AWM) addresses both problems. Its definitions are

$$L(y) = A(N/2), A = \{\underbrace{x_1, x_1, x_1}_{W_1}, x_2, \ldots x_N\} \tag{2.9}$$

$$L(y) = \min(\sum_{i=1}^{N}(W_i * \|x_i - x_j\|)), j \in N \tag{2.10}$$

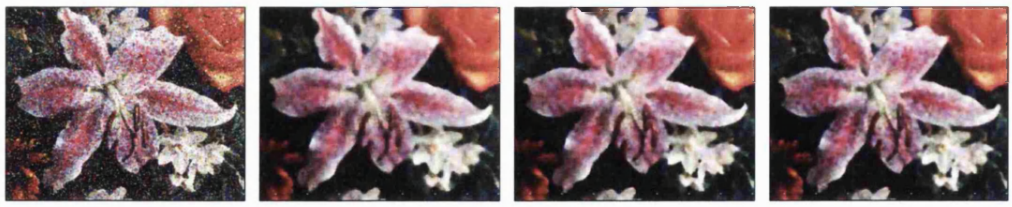
$$L(y) = \min(W_j * \sum_{i=1}^{N}(\|x_i - x_j\|)), j \in N \tag{2.11}$$

The first definition is for greyscale and only takes integer weights, the second definition allows any real values for the weights; in both cases, it is good practice to make them decreasing and positive with distance from the window centre. The third is an alternative definition to 2.10 [20] proposed by Plataniotis, where the weights are applied to the overall metric at each point rather than the individual distances; this also forces weights to be increasing with distance. An alternative to adaptive weights is a weighted combination of filters [107].

The weights definitions can vary, but an example is

$$W_i(x,y) = N - (((N+1)/2 - x)^2 + ((N+1)/2 - y)^2)^{1/2} * (2\delta/\delta_{max}), \quad W_i > 0 \tag{2.12}$$

Clipping any negative weights gives a circular mask for moderate variance regions, with a smaller mask for higher variance. Many authors [93] [108] report this family of filters to give better performance, as seen in figure 2.7.

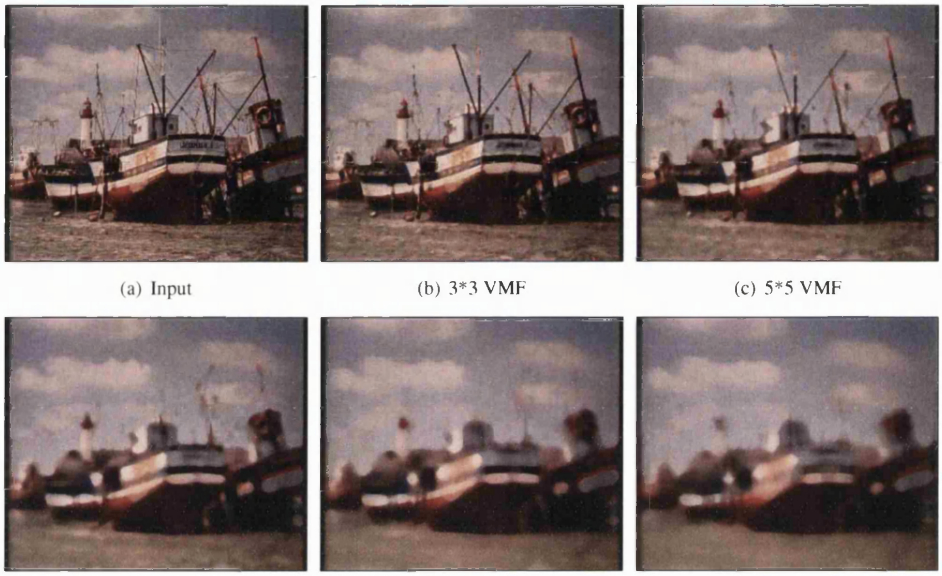


(a) NMSE=15.834*10^{-2} (b) NMSE=3.090*10^{-2} (c) NMSE=2.526*10^{-2} (d) NMSE=2.455*10^{-2}

Figure 2.7: Noise removal: input, normal, adaptive and weighted medians and NMSE

2.4.2 Size Evaluation

The effect of a filter depends on the size and shape of the mask used. Larger mask give better noise removal and simplification but simultaneously force greater distortion [98]. Since the noise performance for all filters is evaluated later, here only the mean and basic vector median filter are considered, with variable mask sizes and L_1 metric.



(a) Input (b) 3*3 VMF (c) 5*5 VMF

(d) 9*9 VMF (e) 13*13 VMF (f) 17*17 VMF

Figure 2.8: Effects of scale on mask filtering

As shown in appendix B.1, the NMSE for median filters hits a minimum at a mask size of 5*5 for Gaussian-related noise and is lower for impulsive noise, while for the mean the opposite is true; in either case, the MCRE is generally decreasing. Visually, figure 2.8 shows that most thin features are lost along with the clear rounding of corners at or above mask sizes of 9*9, so a 5*5 mask size appears reasonable.

2.4.3 Noise Evaluation

All these methods can remove minor detail and noise from digital images. As stated in the introduction to this and the last chapter, noise removal with minimal distortion is one of the main goals of image processing and an important area for nonlinear filters. All those described above except the AWM were evaluated. The image set used here and throughout the thesis is lily, autumn, boats, lenna, baboon and sample1, seen in appendix D.1. All images are of natural content, except sample1, a textured synthetic image. The filter parameters were:

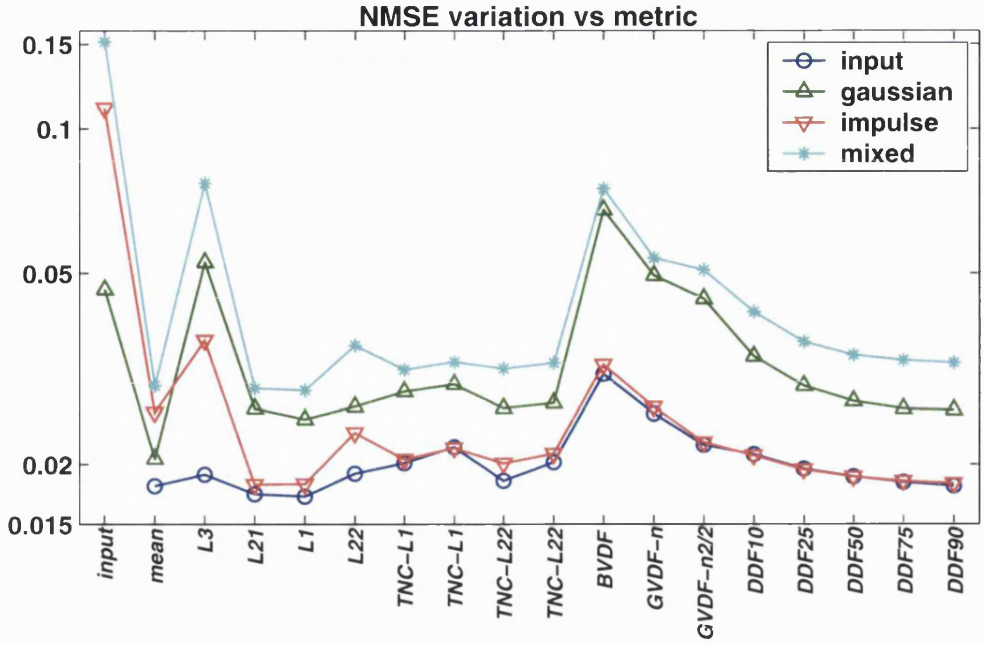


Figure 2.9: Average NMSE for uncorrelated noise

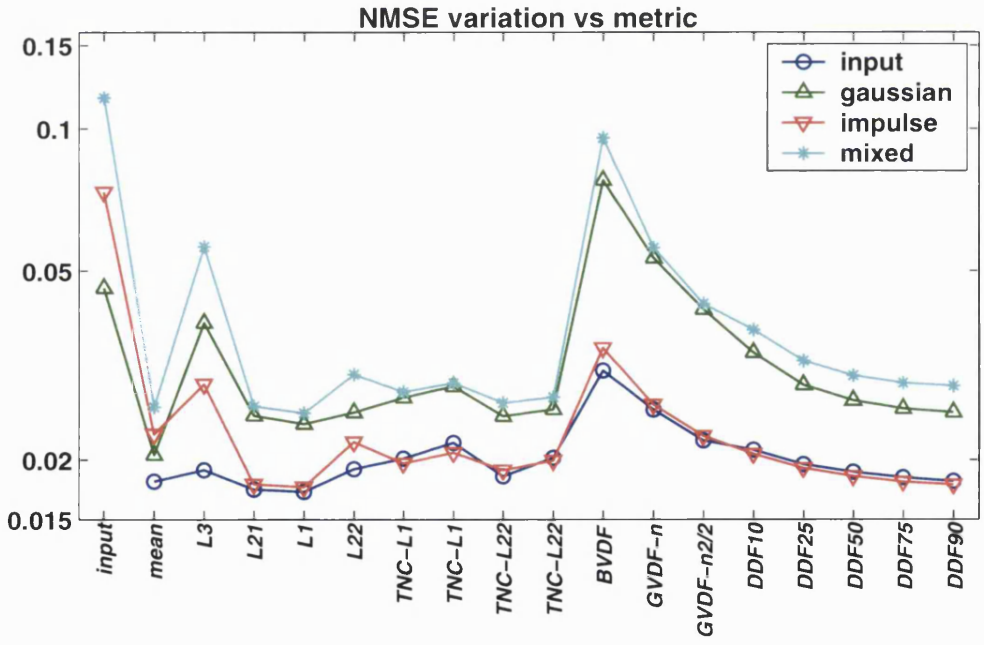


Figure 2.10: Average NMSE for correlated noise

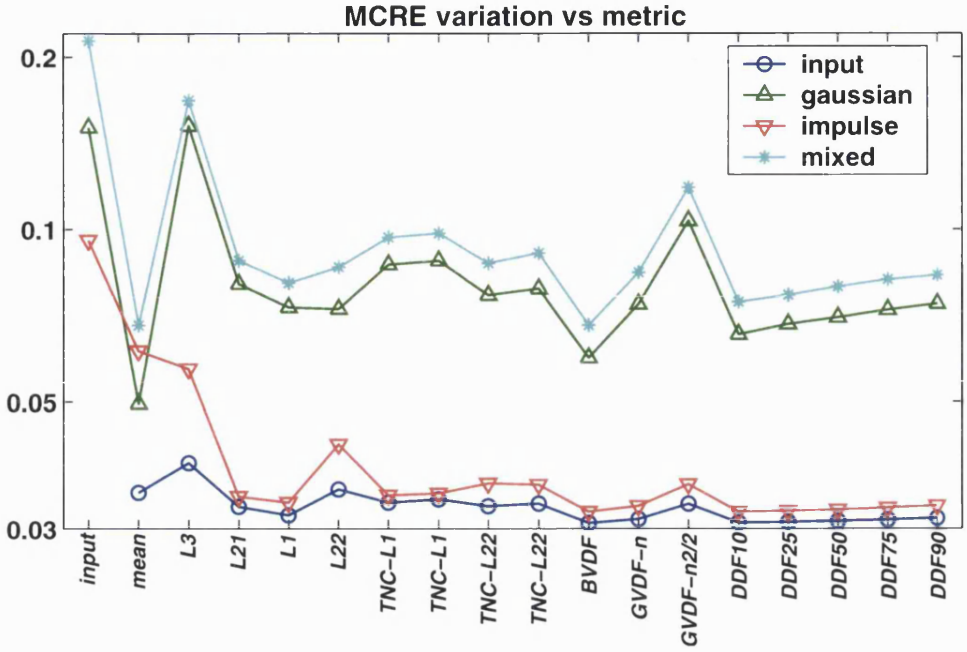


Figure 2.11: Average MCRE for uncorrelated noise

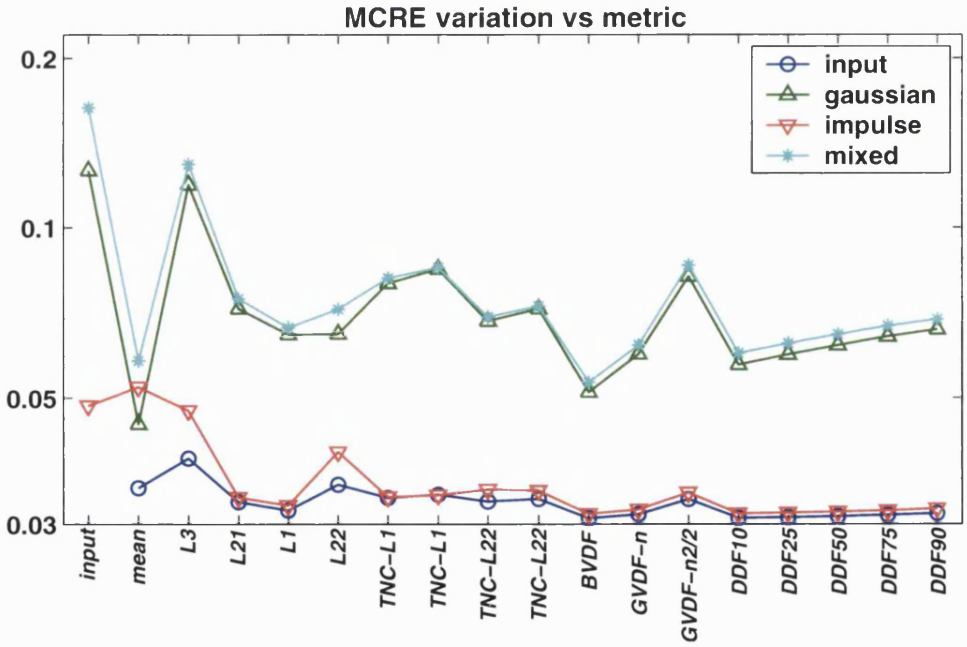


Figure 2.12: Average MCRE for correlated noise

- the VMF with L_1, L_2, L_2^2 and luminance norms
- the GVDF with magnitude averaging over one (giving the BVDF), n, $n^2/2$ samples
- the DDF with 10%,25%,50%,75%,90% of angular median
- the TVMF (or TNC) with L_1, L_2^2 metrics and 1 or 2 iterations

For simplicity, only results of window size 5*5 (giving $n = 5$) and moderate levels Gaussian and impulsive noise are shown for NMSE and MCRE metrics; smaller scales do not give noticeable differences between methods nor the best removal at moderate levels.

The noise types used were no noise (giving input distortion), Gaussian at light (σ^2=100) and moderate (σ^2=1000) levels, impulsive at light (ρ=5%) and moderate (ρ=10%) levels, and mixed noise with σ^2=1000 and ρ=10%. All were generated with uncorrelated and 50% correlated noise models, and results are scaled by a 10^{-2} factor. Test tables are at the appendix B.

Results for the 3 models of moderate/heavy noise show the mean filter is the best for purely Gaussian noise, with NMSE and MCRE 20% lower than any median. This is partly due to the tiling effect of median filters, which is more important in truncated filters: as filter size increases, so does the size of flat regions, a problem when none of the available pixels is a good fit. The distance-based (VMF, TVMF) filters prove best for impulsive noise removal, and for mixed noise the mean still surpasses the other filters, perhaps due to the actual noise levels. In terms of quality, the truncated versions of the filters gave better edge preservation, and proved slightly better when in Gaussian; however, those changes to the image are subjective, which are not measured here.

So considering the NMSE, the best are the L_1 and L_2 vector medians, with the 75% DDF and TVMF near them for mixed noise environments. The TVMFit_1 and 90% DDF are good against Gaussian too, and others are generally negative. Part of this is due to mostly magnitude, and not chroma, appearing in the NMSE. Also the MCRE will decrease for longer as scale increases, often to scales of 15*15 or up to the distortion it generates on the ideal case, generally a low value: this is due to its independence from magnitude.

Another aspect of the evaluation is timing results; these are not shown, and are as follows. The reduced median filters have a complexity of $O(N^2)$, with the K-iteration TVMF following at $O(KN^2)$. The standard medians have a $O(N^3)$ complexity, while the DDF takes the longest, with twice the distance measurements than the $O(N^3)$ standard median.

Metric-wise, the best are those with a magnitude base and which do not rely on single components of the HSI system, e.g. L_3 and BVDF. This is expected, as magnitude plays an important role in the perception of edges and objects; on this point, these two filters get the worst results by concentrating on a component whereas the L_1 median and DDF have the best overall performance. However, the DDF and GVDF achieve acceptable results due to the magnitude weighting. The TVMF surpasses the normal medians in terms of edge enhancement and preservation as shown before in figure 2.6, proving the best for segmentation pre-filtering (by removing more detail and smooth transitions at one scale) and image enhancement.

2.5 Conclusions

Many of the methods have similar performance against noise, up to the scales shown. As seen in figures B.1 and B.2, medians perform clearly better against pure impulsive noise at a mask size of 3*3, whereas for Gaussian or mixed only the NMSE peaks at the 5*5 level whereas the MCRE has no lower limit besides distortion. Above these scales the removal of thin features and the rounding of contours, the main problem, overtakes noise removal.

The best overall filter depends on the task; for noise removal the best are the L_1 metric VMF and the DDF with low/high amounts of the angular metric, depending on whether colour or brightness have a greater importance, although the mean still proves superior against Gaussian noise. The differences are however small, save for the luminance, angular and square Euclidean medians. That reflects the improvement of reduced over conditional ordering in the case of the median and the excess smoothing of the square Euclidean metric.

Finally, the truncated median gives the crispest images, with good noise rejection. That is not reflected by the noise metrics, since the distortion caused by sharpening transitions overcomes any improvement except when other filters have caused severe blurring, e.g. the L_2^2 median or the mean.

Chapter 3

Mathematical Morphology

Initially developed in the 1960's by Serra and more popular from the 1980's, this family of methods derived from rank filtering but based in lattice theory has evolved over time. The first variants of mathematical morphology (MM) were binary operators based on the use of a mask or structuring element (SE), later extended to greyscale images around 1986 by Haralick and Sternberg [109]; this relies on the max() and min() operations or the division in multi-level sets. Finally, tree representations based on connected operators, and therefore independent of the SE, were later developed. Mathematical morphology has some characteristics that are an advantage against median filters, inherent idempotency among others.

The most basic MM operations are erosion and dilation: erosions remove local maxima whereas dilation removes local minima. These are combined to provide openings, closings and sieves, with openings being idempotent and dual with closings. Also thinnings and thickenings are available for non-increasing attributes; granulometries [110] and scale-spaces are also possible by increasing mask sizes for any operation.

All of these are based on the presence of a complete lattice or fully ordered space: all points may be arranged such that $d_0 \leq d_1 \leq d_2 \ldots d_n$ and there exist both a supremum and infimum, the highest possible maxima and minima in the input with a partial ordering relation [111].

The methods studied were designed for greyscale; however they have been often used for colour images by applying a mapping to a fully ordered lattice. The following sections describe the review of these applications. Other morphological applications include skeletonisation, edge-contour extraction and the watershed transform.

3.1 Structuring Element morphology

The first type, developed in the 1960's. As the name indicates, this requires the use of a mask or SE to select the pixels at any location. The main operators, shown in figure 3.1, are:

- erosion $\ominus$: $A \ominus I$=$Min(A(I))$,where A is the SE applied onto image I. Removes the foreground elements that do not match the mask, or takes the minimum of the elements within the mask, while

expanding the background.

- dilation $\oplus$: $A \oplus I$=$Max(A(I))$. Removes the background elements that do not match the mask, or takes the maximum of the elements within the mask, while expanding the foreground, i.e. it is extensive.
- opening $\circ$: $A \circ I$=$A \oplus (A \ominus I)$. Erosion then dilation with the same element, removing the foreground objects. This is idempotent and like erosions, anti-extensive: $A \circ I \subseteq I$ or the output foreground is a subset of the input.
- closing $\bullet$: $A \bullet I$=$A \ominus (A \oplus I)$. Dilation then erosion, or the dual of openings, also being idempotent. Like dilations, it is an extensive operator.

Their key and often present properties are idempotency, causality, duality and extrema division. Extrema division refers to the presence of two extrema types, maxima and minima. Respectively, anti-extensive and extensive operations are available for their removal. Since filters remove the higher or lower extreme, they are robust against impulsive noise, but fare worse for Gaussian, as median type filters [30]. Duality or self-duality is the property of two or one operators, where inverting the input and output images for one filter is identical to using the other or itself respectively; this is derived from the max-min classification.

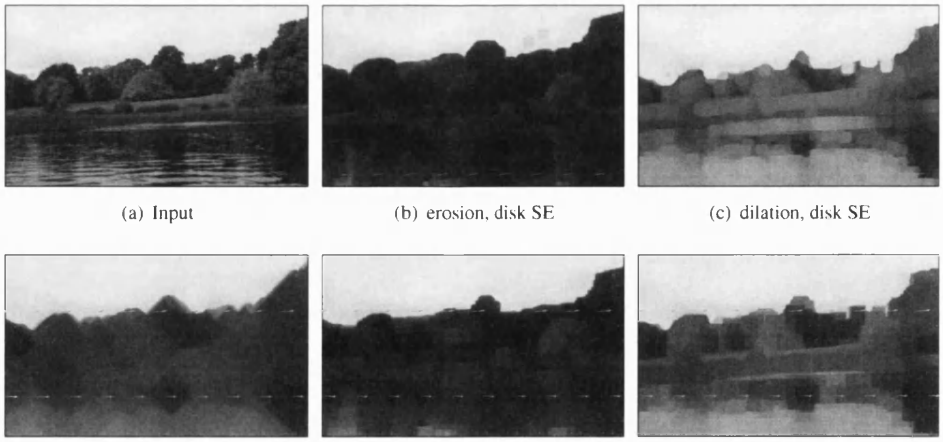
Causality, when referring to multiscale representations, is the absence of new features at higher scales and the dependency only on finer ones, i.e. causal operators working on the initial image or a medium-scale processed version give the same results at higher scales; all the operators above are causal, although dilations and erosions will need a different-sized SE since they are not idempotent. Idempotency, the most important, implies that using a dual or self-dual operator (openings, closings, sieves) on an image already treated once by that operator will have no effect, unlike the median, where $\frac{n}{2}$ passes are required for idempotency in the n-length median or just one for the recursive median [101] [112]. From a practical point of view, applying such an operator on its own or as part of a cascade of filters on a processed image will have no effect unless it's been corrupted.

Combining the prior options allows for:

- proper openings and closings [30], or $Max(I, (A \circ (A \bullet (A \circ I))))$ and $Min(I, (A \bullet (A \circ (A \bullet I))))$
- the automedian or morphological centre [30] from the proper open-close
- openings (and closings) by reconstruction $Max(I, (A \circ I))$, leading to top-hat operators.
- the sieves, taking openings (ϕ_λ), closings (ψ_λ) or both (ASF) at increasing scales [113] [114], as $\phi_\lambda = A_\lambda \circ (A_{\lambda-1} \circ (\ldots (A_2 \circ (A_1 \circ I))))$

Other filters are the local monotonic (lomo) filters as suggested by Acton [115], so-called soft [116] and adaptive [117] morphology.

A common feature to all methods, clearer in figures 3.1b-f, is that the shape of the SE will superimpose on the output; this removes parts not shaped like it but superimposes its pattern on the remaining elements. An alternative to this is combining different masking elements [41], whether it be in orientation [118] or in shape, leading to area and tree morphology.



(a) Input (b) erosion, disk SE (c) dilation, disk SE

(d) ASF, diamond SE (e) opening, square SE (f) closing, square SE

Figure 3.1: SE morphology example

3.2 Area morphology

This type of filtering, also known as graph or tree morphology, creates a hierarchy or tree of the image, giving an ordering between the smallest regions as leaf nodes at the bottom of the tree and their parent nodes, with a root node at the top of the structure [119]. Once the tree is created, filtering follows by pruning, or merging the leaf nodes to their parents, until a criterion (area, complexity, ...) is achieved. Linear scale spaces [120] are older and based on the diffusion equation, both linear and nonlinear ones falling in another category to morphological ones [121] [122] [123].

There are two main advantages over SE morphology: connectivity and tree creation. Connectivity is a basic concept of graph theory: if there is a path with any number of segments between two given nodes in a graph, they are connected. An image can then be considered as a graph composed of vertices (pixels) and connecting segments or edges. The possible segments depend on the local connectivity between adjacent nodes, often being restricted to the 4 or 8 surrounding pixels for images and 6, 18 or 26 voxels for volumes, but second-order connectivities also exist [124]. This gives rise to flat zones as connected components, i.e. fully connected subgraphs. A connected operator can be defined as follows [124] [125]:

- Two points p_1 and p_2 in image I belong to a flat zone Z_i if they have the same value and are connected by a valid path. Their union gives an image partition $P(I) = \bigcup_i Z_i$, where $Z_i \cap Z_j = \emptyset \forall i, j$, so flat zones cannot overlap.

- An image partition $P_A(I) = \bigcup_i Z_i^A$ is finer than another partition $P_B(I)$ if the flat zones P_A are subsets of those in P_B.

$$\forall i, p_1, p_2 \quad p_1, p_2 \in Z_i^A \Longrightarrow \exists j \text{ where } p_1, p_2 \in Z_j^B \tag{3.1}$$

- An operator $\Psi()$ is connected if for any image I $P(I)$ is finer than $P(\Psi(I))$.

The most common connected operators [110] [126] are equivalent to their SE analogues:

- The area opening or the supremum of all possible SE openings: $\gamma^a_\lambda(I) = \vee_{B \in A_\lambda}(I \circ B)$
- The area closing or the infimum of all possible SE closings: $\phi^a_\lambda(I) = \wedge_{B \in A_\lambda}(I \bullet B)$
- The alternating sieve $ASF(I)_\lambda = \gamma_\lambda\phi_\lambda(\gamma_{\lambda-1}\phi_{\lambda-1}(\ldots\gamma_1\phi_1(I)))$

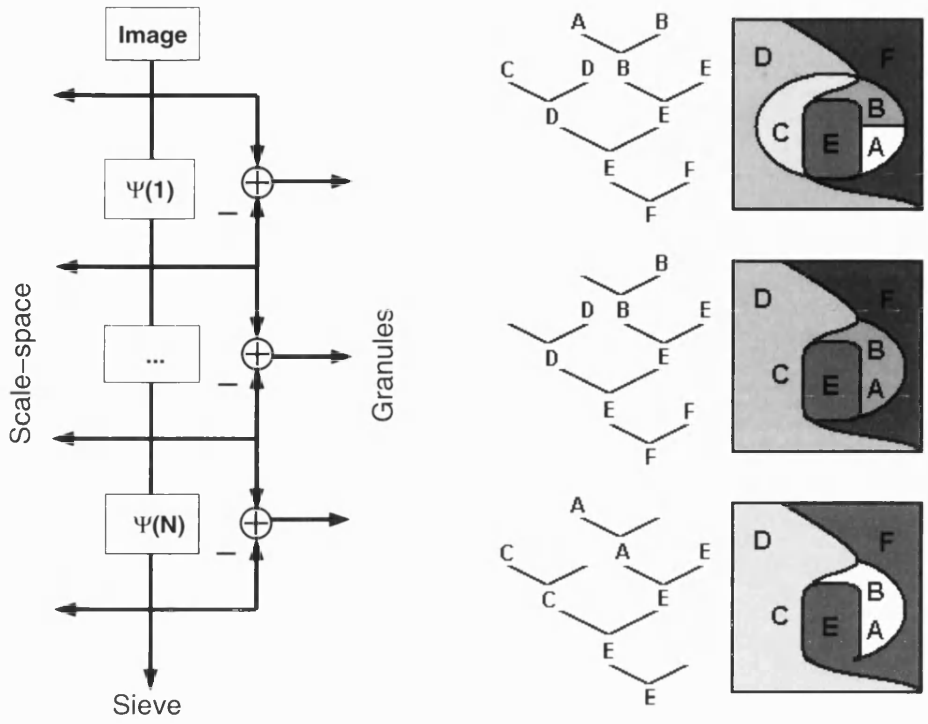
Connected operators are linked to general region merging methods by working on an image and its partition at once [55]. Like in SE morphology, $\gamma^a_\alpha(I) = \gamma^a_\alpha(\gamma^a_\beta(I)), \forall\alpha \subset \beta$; but unlike it, α and β are areas instead of SE. This implies that all connected pixels with a valid path, or equal values in this case, are treated as one set with common attributes like area; this also makes it independent of the element chosen since there is none. That leads to the fact shapes and contours are not created or altered unlike in mask-based filtering, which is the property of strong causality [127]. As seen in figure 3.2a, these properties allow for multi-scale decomposition, not only as the simplified outputs at each scale, giving a scale-space as medians or diffusion methods do, but as a granulometry from the changes between successive scales (granules) since well-defined object disappear at a particular scale rather than a broad range thanks to shape preservation (i.e. strong causality).

Another advantage of connected operators is the ease to apply other attributes. This includes increasing attributes where region growth always gives a greater or equal metric, like area and power, and non-increasing attributes, like most shape constraints. Increasing attributes allow the use of openings and closings, whereas non-increasing ones give rise to thinnings and thickenings.

Attribute sieves create a tree structure due to being connected operators, i.e. not splitting flat zones, and their merging order. A tree is a graph without any cycles, where removing a segment will split the tree. The most common type used is the ordered rooted tree, with an order among nodes with the same parent and all paths pointing towards the root, from the leafs (original regions) to the root.

There are different types of trees: partition, scale/sieve, max-/min- trees, etc. Here only sieve and Max-/Min-trees are considered [125]. The Max-tree considers the local maxima as the leaf nodes, merging them to the nearest region with lower value, whereas its dual the Min-tree does the reverse: that is, $MaxTree(f_{(x,y)}) = -MinTree(-f_{(x,y)})$. The sieve tree, or scale tree, considers the smallest extreme regions as the leaf nodes, both maxima and minima; the ordering can affect the results, but using conventions as alternating filters can reduce that problem [128].

The use of trees means attribute sieving reduces to pruning of the tree, with some constraints (ordered trees, attribute changes) on certain applications. This is readily extended to most attributes, even non-increasing ones [110] [56]. Also a semantic comparison is possible among different images, considering the higher branches as separate objects [129]. In addition, scale-spaces obtained by tree processing obey strong causality, derived from non-overlapping regions (nodes) being subsets of their parent. Finally, other forms of tree processing exist and are investigated in section 7.4; as shown in figure 3.2b, these can process nodes besides the smallest leafs as in sieves, such as tree collapsing where non-leafs are removed, and provide face values other than those in the sieve - the darkest in this example, since Max-Tree extrema/leafs are brighter.



(a) Relationship between sieving processes

(b) Max-Tree construction, sieving and collapsing

Figure 3.2: Area morphology advantages

There are multiple implementations for openings or closings [130]; alternating sieves are often produced by multiple scans, what proves inefficient at large scales.

- Priority queue: the earliest method, based on a pixel queue. Simple but slow if nested maxima are present.
- Max-Tree [56]: more flexible in terms of post-processing than other methods. Independent of scale, but slow on startup as the image must be processed to the top scale.
- Tarjan's union method [130]: the fastest, almost independent of sieving scale. Based on the union-find method.

The general algorithm used for openings follows below:

1. Identification of extrema (maxima) regions.
2. Merge all scale 1 extrema regions to the nearest neighbour.
3. Repeat step 2 with increasing scales until the scale desired.
4. If using a tree, remove all nodes up to the desired scale.

Note that for openings or closings it is possible to merge straight to the final scale. The pixel queue method keeps an ordered list of neighbours to merge, rebuilding it for each extrema. For the Max-Tree approach, all identical values connected to a given extrema often become one node or neighbour set, relying on histograms and/or recursive flooding before pruning the resulting tree. Tarjan's algorithm deals with multiple sets rather than individual pixels as the other two, starting from the highest greyscale value for openings (implicitly maxima): if the parent node value of an active set is still extreme, the union operator merges it to the nearest set (and future parent), else it is deactivated and kept.

3.3 Colour morphology operators

The extension of morphological methods to colour is not straightforward due to the problems with supremum/ infimum definitions, as seen in chapter 2. Here, only marginal, conditional and projection ordering approaches were studied in depth.

The main reason is that both other types of ordering (partial, reduced) give a measure of extremeness, not of magnitude; that ordering must be recalculated after each merge. In the case of reduced orderings with a reference to neighbours, higher values relate to outliers, without a specific direction: this makes the notion of supremum/ infimum impossible without further refinement, such as finding outliers then using luminance or distance to a reference point [131] or the local population for further ordering, and can produce multiple adjacent extrema; for partial ordering this is worsened by the lack of ordering within each subgroup. However, a reduced, projection ordering, like luminance, dot product or restricted hue, only relies on the actual point and as in figure 3.3a behaves like the conditional case, avoiding all problems other than the adjacent extrema.

Both these ordering schemes can also induce extrema creation since the many-to-one transforms imply that the best matching (regardless of outlierness) and closest metrical neighbour may not be the same, giving 'contours' around the original edges as in figure 3.3b, or minor extrema when constrained by luminance above or below the original value as in 3.3c, which will be detected when the metric is updated. In their favour, however, there are also examples which define the extrema locally [7], considering only those pixels forming a local group with a clear centre when dealing with hue; Gillet [57] proposes fuzzy erosions/ dilations on the image distribution before normal clustering, and Salembier [132] exposes a partial morphological reconstruction method for edge restoration. Morphology-like schemes for the HSI ordering of RGB data were not studied in depth; as seen, hue maxima or range are often selected with other techniques [133].

In addition, many colour methods in previous literature have involved marginal or conditional ordering on the input. An example is SE hue morphology [131]; this allows to consider objects by colour, what implies a tone and its complementary are the supremum and infimum for the image if brightness and saturation remain constant; this is analogue to the concept of area MM [133], where a hue range without visible discontinuities (e.g. no pure green hue for a 120° to -240° range) is selected from global histograms. Luminance ordering, or addition of all/several channels, is another trivial example of conditional ordering; as noted above, this only involves the actual region, and hence will not generate extrema.

A number of alternative metrics also involve a conditional component. Zaharescu et al. [134] [135]

(a) blue hue erosion (b) L_2 R-ordering dilation (c) L_2 R-ordering clipped dilation

Figure 3.3: Colour morphology SE examples on 'autumn' image; 7*7 mask

propose a new geometric metric based on the triangle delimited by the RGB values; however, this proves more similar to the conditional ordering schemes, where the k-ratio described is more oriented to the saturation component and the area and perimeter of said triangle have a link to intensity. Chanussot [136] also provides for a fully ordered scheme based on bit interleaving: although promising, this was not implemented due to the potential bias in favour of the first channel being used, and hue-based results for RGB vectors as in [131]. Besides, problems in RGB are more important, and can be extended to any number of dimensions.

In brief, area operators were implemented only for conditional, projection or marginal methods, avoiding new extrema, while mask-based operators were also allowed reduced distance ordering. The algorithm used was the Max-tree but with a priority queue approach, which is unlike that in chapter 4. One of the initial image extrema is merged to neighbouring sets at lower extremeness levels until it ceases to be an outlier, moving to the next region in the list until the tree is built; sieving, or pruning, is done as usual. This allowed the construction of Max and Min-trees, but not Sieve-trees due to order-related artifacts.

Regardless of all these problems, some trials were carried out in colour SE morphology, starting with marginal morphology and sieves, then progressing to conditional and reduced metrics. These included point measures such as hue [131] or luminance-based morphology, and distance-based ones, using a centroid-based scheme similar to the median. This gave some interesting results for hue-based region retrieval (see above), but otherwise proved inconvenient. The results were unsatisfactory and therefore discarded, due to the SE median-like effect of rounding at the corners, the problems of combining erosions and especially dilations, and the lack of a single metric giving a complete image ordering [137].

3.3.1 Noise Evaluation

Evaluation was done in terms of noise removal performance for the morphological area opening, closing and their combination, using the NMSE and MCRE. As explained before, reduced metrics were ignored as they induce severe distortion and new extrema even with constraints, which foils the assumption that sieves only destroy outliers. The metrics studied were:

- marginal ordering on the RGB image.
- conditional ordering with only 1 component. These include the individual components of the HSL space together with vector angle and luminance.

- conditional ordering with multiple components. These include a blend of the above, with luminance and chromaticity components united.

Metrics used are: L_g=greyscale luminance, L_L=HSL brightness, L_{LH}=brightness and HSL saturation, L_{gH}=luminance and HSI saturation-distinct, L_M=marginal.

Type		2	5	10	50	100	500	1000
O	L_g	0.2289673	0.7081862	1.236699	3.444336	4.858003	13.65378	18.38492
	L_L	0.2548608	0.7623280	1.360897	3.703669	5.395471	13.22406	18.43849
	L_{LH}	0.2502338	0.7542049	1.353995	3.707372	5.327965	13.36489	18.33422
	L_{gH}	0.2238060	0.7037740	1.225244	3.523486	4.966272	13.75131	18.23792
	L_M	0.1481625	0.5435987	1.027562	3.081754	4.402588	12.09510	16.89102
C	L_g	0.2343381	0.7832241	1.487749	3.422603	4.910715	11.64365	14.77461
	L_L	0.2423257	0.8661453	1.568548	3.711685	5.224077	11.81635	15.16737
	L_{LH}	0.2365940	0.8617443	1.562953	3.726108	5.254663	11.76073	15.21987
	L_{gH}	0.2363822	0.7892491	1.493264	3.436241	4.964305	11.68432	14.90803
	L_M	0.1806066	0.6766849	1.308611	3.015096	4.338028	9.94037	12.36048
OC	L_g	0.4594739	1.478746	2.684491	6.661072	9.436374	24.22300	31.88992
	L_L	0.4902684	1.609868	2.883996	7.201319	10.30457	23.88989	31.69906
	L_{LH}	0.4820668	1.598157	2.875440	7.216395	10.20546	24.28036	31.54833
	L_{gH}	0.4561612	1.476064	2.683322	6.757046	9.543231	24.70570	31.67366
	L_M	0.3226678	1.200324	2.292520	5.888634	8.373806	20.96367	27.47266

Table 3.1a: Average NMSE vs scale on uncorrupted input/10^{-3}

Type		2	5	10	50	100	500	1000
O	L_g	1.712258	3.758226	5.247260	8.784973	10.40851	17.42183	21.08560
	L_L	1.828389	3.960395	5.616651	9.190267	11.27736	17.02875	21.22632
	L_{LH}	1.718474	3.804719	5.467497	9.103854	11.02262	16.89920	20.85596
	L_{gH}	1.648281	3.702187	5.170772	8.857692	10.58317	17.65848	20.37324
	L_M	1.427514	3.392128	5.020582	8.630499	10.10441	15.39909	18.86439
C	L_g	2.108776	4.543742	6.429446	10.11086	12.21270	18.35309	22.48591
	L_L	2.089892	4.777485	6.637953	10.54227	12.15426	17.95744	20.78243
	L_{LH}	1.933203	4.605800	6.462212	10.49287	12.27665	17.35584	20.47513
	L_{gH}	2.048485	4.456253	6.325420	10.12467	11.98449	18.10705	23.54819
	L_M	1.634997	3.801786	5.392917	8.644254	10.31548	15.05740	18.34124
OC	L_g	3.774069	8.147362	11.39486	18.24912	21.95422	33.93296	41.97509
	L_L	3.832616	8.536879	11.95084	19.00969	23.04413	33.04249	39.08569
	L_{LH}	3.594966	8.263415	11.69181	19.02331	22.49965	33.69399	38.61039
	L_{gH}	3.652886	8.019354	11.27876	18.50249	21.97322	34.39437	41.54316
	L_M	2.976434	6.935218	9.951154	16.11550	18.95388	27.83662	33.22015

Table 3.1b: Average MCRE vs scale on uncorrupted input/10^{-3}

Max and Min-tree approaches have improved success due to the lack of an SE. However, efforts at combined max-min trees have proven difficult, what gives the problem of isolated extrema of one or other kind after sieving. Young [138] shows there is little difference between using AF and ASF, at least for low scales. Due to this, the Max and Min-trees and the alternating filter from both are shown. Although prior sieve trees have been created, this option was discarded due to the higher complexity and variable

Type		none	$\delta^2 = 100$	$\delta^2 = 1000$	$\rho = 5\%$	$\rho = 10\%$	mixed
input		0	4.653768	41.12448	52.37775	103.2408	127.8278
O	L_g	1.236699	5.581248	38.51805	29.67491	58.76998	89.86186
	L_L	1.360897	5.730644	38.99321	29.50799	58.00644	90.28469
	L_{LH}	1.353995	5.710521	38.95257	29.50900	57.94959	90.09644
	L_{gH}	1.225244	5.562004	38.37519	29.73767	58.87379	89.84923
	L_M	1.027562	4.251932	26.59151	27.10861	52.13394	69.84505
C	L_g	1.487749	5.829496	38.91768	31.32946	61.61954	93.00011
	L_L	1.568548	5.997837	39.34999	31.44719	61.98531	93.01556
	L_{LH}	1.562953	5.996958	39.36147	31.45689	62.03396	93.04449
	L_{gH}	1.493264	5.851962	38.95922	31.39302	61.75539	93.14636
	L_M	1.308611	4.464246	26.93128	28.82422	56.13267	73.49775
OC	L_g	2.684491	6.707993	36.21989	8.376068	16.99667	54.80399
	L_L	2.883996	7.002842	37.18687	8.470517	16.54935	55.52967
	L_{LH}	2.875440	6.991379	37.14500	8.478488	16.50539	55.18190
	L_{gH}	2.683322	6.696451	36.09715	8.515555	17.30064	55.05037
	L_M	2.292520	4.034413	12.83250	3.492357	4.98300	16.10963

Table 3.2a: Average NMSE for tree morphology metrics vs noise/10^{-3}

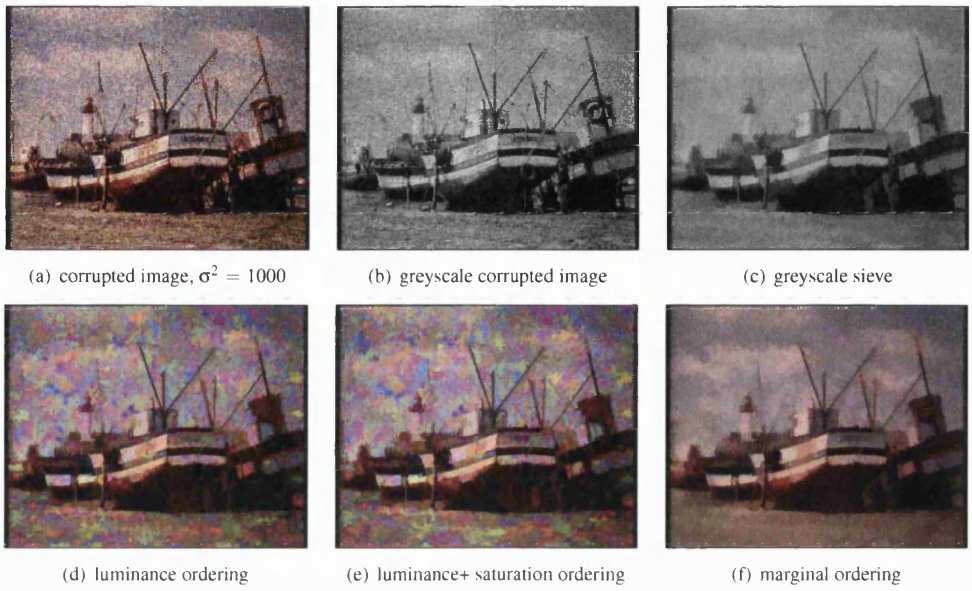
Type		none	$\delta^2 = 100$	$\delta^2 = 1000$	$\rho = 5\%$	$\rho = 10\%$	mixed
input		0	54.00727	149.4230	48.70511	93.58154	201.0940
O	L_g	5.247260	58.08998	159.0622	39.60731	72.37378	196.4280
	L_L	5.616651	57.95564	157.8264	39.60189	71.29226	195.8210
	L_{gH}	5.467497	57.93150	157.7779	39.50532	71.13183	195.7469
	L_{LH}	5.170772	58.05130	158.8831	39.65702	72.43634	196.4698
	L_M	5.020582	47.05907	127.7402	38.78908	70.96444	167.6949
C	L_g	6.429446	53.73644	139.0964	28.62986	50.30648	162.8029
	L_L	6.637953	54.28151	142.4819	29.26470	51.46918	166.9465
	L_{gH}	6.462212	54.45553	142.4443	29.24037	51.49460	166.9197
	L_{LH}	6.325420	54.21837	139.2383	28.70389	50.51928	163.0943
	L_M	5.392917	37.59882	88.54605	25.25175	43.55024	109.8398
OC	L_g	11.39486	58.22981	150.1948	19.07445	28.80913	160.5820
	L_L	11.95084	58.42209	151.8157	19.78660	28.59678	164.2400
	L_{gH}	11.69181	58.58510	151.7192	19.71694	28.53156	164.0759
	L_{LH}	11.27876	58.41359	150.1685	19.27669	29.19288	160.8616
	L_M	9.951154	32.30878	71.94952	14.49491	19.03719	78.01655

Table 3.2b: Average MCRE for tree morphology metrics vs noise/10^{-3}

merging order. Salembier [56] shows an alternative, using a separate algorithm to create the tree; this is considered in latter chapters.

The named sieves and metrics were evaluated on the set of images 'boats', 'autumn', 'lily' and 'sample1'. For clarity, only the average results are shown, with distortion versus scale for only the input and metric vs noise source for all cases. The effect of noise is shown only at scale 10. The reason for this choice is both corrupted NMSE and MCRE achieve their minimum values at scales of 5,10 or 20 over most images with little or no improvement beyond, what can be justified by the low-scale properties of noise. As noted above, all the metrics shown have a luminance/brightness base due to the negative results

of hue and saturation, which mimic those for the BVDF (see vector median evaluation). Figure 3.4 shows the differences when using an AOC sieve on 'boats' image to area 100 compared to greyscale.



(a) corrupted image, $\sigma^2 = 1000$ (b) greyscale corrupted image (c) greyscale sieve

(d) luminance ordering (e) luminance+ saturation ordering (f) marginal ordering

Figure 3.4: Graph morphology results on 'boats' image

Under all conditions, marginal ordering, especially in alternating filters, surpasses all other conditional ordering methods. Qualitatively, this is clear in figures 3.4d-f, where local hues of conditional schemes are inconsistent while the quality of marginal ordering is analogue to that of greyscale methods, making it useful for both simplification (and scale-space segmentation) and denoising. This reflects itself for Gaussian noise in terms of MCRE, where it achieves scores 50% lower than the other methods in table 3.2b, possibly due to the creation of new colours nearer to the unaffected average, inducing less distortion. The other schemes achieve different scores depending on the noise type, with scores differing by 2%-8% : the luminance-based orderings are best for Gaussian-related noise due to all channels being considered, whereas those based on brightness are better for pure impulsive noise due to the fact brightness takes the maxima and minima, either of which contain outliers for impulsive noise; scale and distortion-wise (see table 3.1a) luminance orderings are marginally better than the former except at high scales. The AF gives the best overall results as expected; however, only closings lower the MCRE for Gaussian noise, probably due to most of the colour information and noise being in the higher end of saturation and brightness, and thus enhanced by openings.

All other metrics studied (saturation, hue or angle with optional luminance support) proved far worse, due to their inability to consider luminance with the required importance like in the BVDF median. The results are similar for SE morphology; however, the use of the area attribute and of a region-wise representation makes the area morphology better, at least in terms of contour preservation.

3.4 Conclusions

Graph morphology is a good greyscale technique in terms of noise removal like the median, yet achieves idempotency in one cycle and has no associated contour distortion, unlike the recursive and normal median and mask-based morphology. However, the results in figure 3.4 show that simple extension to colour spaces is inefficient, due to the vector data not admitting a full ordering scheme which represents all outliers.

Overall, vector medians are far better at noise removal despite edge distortion (absent in area morphology) due to the ordering dilemma, i.e. reduced ordering proves the best for noise removal and suits medians. However, except for mappings with one input, e.g. luminance, this proves unsuitable even for area closings. In such cases, determining the local population size (and the use of regions, points or region areas when finding a metric) and finding the nearest metric value to a maximum when opening (the most different adjacent region, i.e. another maximum) gives effects like those in figure 3.3b. All these factors lead to the conclusion that no full ordering scheme can achieve a better performance than marginal ordering morphology, which again proves the fact mentioned by Zaharescu [134] for the difficulty of ordering definitions for mathematical morphology, so specific colour methods are needed [135].

Chapter 4

Colour sieving methods

Greyscale sieves, and in particular attribute sieves, have a set of desirable properties in terms of noise removal and image analysis. Sieves also give the option of more general tree post-processing, besides granulometries and scale spaces.

Granulometries give an insight to the size structure of an image by finding the granules or differences between successive scales; that also allows for shape granulometries [139] with different SE sieves. Scale spaces are the collection of sieved images across a range of scales, giving a multi-segmentation in the process. However, tree post-processing has the most potential for image analysis, since branches at different attribute values or depths can then be associated to actual objects, allowing for semantic comparisons among different trees and better image segmentation.

However, as exposed in prior chapters, colour in mathematical morphology is rather problematic; simple marginal processing generally gives negative results (new values, edge smearing, biasing towards the space axes) and most other times this involves mapping to greyscale or the local population/ mask. Some methods give morphology-based or aided segmentations, by clustering [5] or reordering [10] [4] the vector data. Watersheds are an exception to these problems, but they rely on a scalar gradient in most cases [140]. All of these give a set of regions or just borders, not an actual image, rarely being suitable for denoising or low scale processes. Morphological sieves can do both, and also enforce strong causality.

4.1 General problems

Colour spaces differ from greyscale in some points. This especially affects segmentation and morphology. The main one is extrema definition, which has no unambiguous definition unlike in greyscale. This is due to the vector nature of the data which makes full ordering more complex or less representative; this also affect the definition of outliers, as marginal or conditional ordering will ignore outliers in other planes, especially in the hue component; however, it is possible to order the hue data by considering a specific hue as the minimum [131], using circular operators [7] or a full lattice [141]. Besides, partial and reduced orderings give a measure of outlierness yet exclude the definition of supremum/ infimum points,

being many-to-one transforms. That can potentially give regions both adjacent and extreme, something counterintuitive to the traditional greyscale approach.

Another difference is in selecting the closest neighbour, which is more complex. As noted by Salembier [56], the merging criterion and the merging order (extremeness) can be treated as separate concepts, whereas most classic methods consider them as the common segmentation criterion. Many options have been considered, with flooding methods favouring the latter approach [140] [5] together with conditional/reduced orderings for best neighbour selection [141]. Since the merging criterion and order are not necessarily the same, there is room for extrema generation, unlike in most greyscale methods. As shown further on, some merging criteria are more prone to this problem.

Finally, the presence of multiple colour spaces means outlier detection is space-dependent. As detailed in the introduction some spaces are better than others, not including the use of principal component analysis (PCA) for information gathering; greyscale offers no such problem. Also noise models and other features, explained before, are different due to interchannel correlation.

4.2 Colour morphological spaces

The use of morphology sieves in greyscale is motivated partly by the presence of scale-spaces; this is a powerful tool for segmentation and detail removal, with the added bonus of strong causality for morphological scale-spaces [127]. Until recently, attempts have been few due to the lack of a proper extrema reference; a common way around in colour morphology is to separate colour and luminance processing [26] [9] before combining their results. In a few cases (the CCS [142], VAMS [143], morphological circle SE [7], vectorial SE [137], cluster ordering openings [144]) there has been an actual colour sieving operation.

Recently two methods for colour sieving have been developed: the VAMS and the CCS. These two approaches provide outlier definition in terms of the surroundings, i.e. the extremeness metric depends on the adjacent regions [10], unlike full ordering schemes; this is novel in their use to morphology, where fuzzy morphology [57] is already known.

For greyscale sieves the structure is:

1. Identify extrema regions
2. Merge all scale 1 extrema regions to the nearest neighbour
3. Repeat all steps with increasing scales (only step 2 for openings or closings) until 1 region is left.

The methods studied overpass the greyscale adaptations studied before as merging order and criterion are independent: removed items take the value of the nearest neighbour in the vector space, not in the extremeness scale, solving the problems of reduced ordering morphology detailed in section 3.3. The priority queue algorithm used for the colour methods enforces sieving by scale, making it a better option for Sieve-trees than the method used in chapter 3. In addition, the metric for extremeness is not a conditional hue or luminance-based, but a reduced one: the difference between regions, not the region's value, determines the metric.

Both operators are also region-based, not pixel-based, and they are a form of attribute morphology, being idempotent connected operators. Due to this, strong causality (no new boundaries or regions are created with increasing scales) is accomplished. From it, scale-spaces and sieve trees are easily derived.

The main drawback of this reduced type of ordering is that adjacent regions can both be extreme: this is easier for the CCS since only 2 states exist. In the VAMS this forces considering flat intensity plateaus as extreme, since a higher degree of neighbourhood would be needed to remove doubts. Also extrema may be destroyed orcreated at any stage in colour sieves.

4.2.1 Colour sieving Algorithm

Since extrema can appear in colour, the greyscale segmentation scheme must be updated:

1. Identify extrema regions
2. Merge all scale 1 extrema regions to the nearest neighbour
3. Repeat steps 1 and 2 until no extrema are found up to the current scale.
4. Repeat all steps with increasing scales until 1 region is left.

The 3rd step ensures idempotency despite extrema generation. That is possible because this reduced ordering depends on the current neighbours, and because the merging criterion (best fit neighbour) is another distance metric (reduced ordering), taking the minimum value to a region. This fact will alter the extremeness of any connected regions, effectively creating new extrema around the old ones.

Two types of outliers are also mentioned in this section: clear outliers (which correspond to noise or points which are clearly uncommon or on the outskirts of a cluster when clustering in the colour space) and false outliers (points which do not have an extreme or rare value but are treated as such). Extrema definitions are different and often clashing in both methods, yet sometimes overlap. An example of clear outliers is those in the HLS space with little resemblance to the neighbours, e.g. high amplitude noise and highlights/shadows. For other outliers, these include items resembling their neighbours yet identified as outliers due to their size, shape, neighbourhood size and other value-independent properties; note that the VAMS takes edge points as outliers, what can be deemed correct when considering transitions between data clusters as out of place.

4.2.2 Convex Colour Sieve (CCS)

The CCS [142] [145] involves getting the spatial outliers with a limited partial ordering, relating to the greyscale definition of outliers (supremum and infimum). This is done with the local convex hull of a region vector and all neighbouring ones; if the region's vector is on the hull, then it is considered extreme and merged, else it is ignored. The convex hull of the set X with n points will contain all points given by

$$H(X) = \left\{ \sum_{j=1}^{n} \beta_j x_j \quad : x_j \in X, 0 <= \beta_j <= 1, \sum_{j=1}^{n} \beta_j = 1 \right\} \tag{4.1}$$

taking real weights β_j. This is the minimal convex set of all point combinations, or the smallest containing space without caving in at any point: any point interpolated or in between others falls in the hull. The points h_x on the hull corners are those where $h_x \notin \{\sum_{j=1}^{n} \beta_j x_j : \beta_j < 1 \forall j\}$ or cannot be interpolated from others; points on the edges or faces can only be obtained from collinear or coplanar corner points.

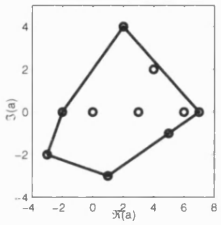
Since the hull is scaling and rotation invariant, most linear transformations can be used on the hull, and the same outliers are found in all cases. The extrema detected involve clear outliers (both noise and greyscale-equivalent extrema, which are on the hull) together with most other points for simpler hulls; the reason being a n-D hull needs at least (n+1) points unless it is degenerate, or 4 points for colour. Figure 4.1 gives the global hull extrema and the convex hull in bold for a vector field; all values except (0+0i), (3+0i), (6+0i), (4+2i) are on the hull so regions 1,2,5,6,8 and 11 are in the hull and therefore not outliers of the global hull. Note regions 1, 6 and 8 have the same value, region 3 (5-1i) is on the border of the hull and the CCS takes the local, not global, hull.

0+0i	4+2i	4+2i	5−1i	−3−2i
0+0i	2+4i	5−1i	5−1i	6+0i
2+4i	2+4i	5−1i	7+0i	6+0i
0+0i	2+4i	1−3i	3+0i	0+0i
0+0i	0+0i	1−3i	0+0i	−2+0i

(a) input

1	2	2	**3**	**9**
1	**4**	**3**	**3**	5
4	**4**	**3**	**10**	5
6	**4**	**7**	11	**8**
6	6	**7**	**8**	**12**

(b) region map



(c) hull global

Figure 4.1: CCS example, 8nn connectivity

The result is a high proportion of extrema throughout the entire sieve, evenly scattered at all scales. Since all extrema are considered equally important, only maxima are defined, so just openings are available. Sieving will sharpen most boundaries (due to the limited hull size), remove noise and simplify regions up to the limit scale - merging is not bound by the image contents, as extrema numbers remain high and evenly scattered. This method reduces to a normal sieve in greyscale inputs, since points then map to a line, combining openings and closings in a random order, unlike M or N sieves.

This implementation differs from Gibson et al. [142] in the treatment of degenerate cases; the convex hull algorithm is QHull [146]. Since QHull does not consider the degenerate cases and most hulls with more than 3 points are non-degenerate, only 2 and 3-point cases are dealt with; the latter considers points as extreme if the distance to its neighbours is less than the distance between those neighbours, what effectively removes some of those outliers. All other options are ignored by 'joggling' the input and removing coplanar cases. This does not affect results highly; besides, the alternative would be a 'tolerance factor' to account for the limited precision of the space used in order to consider collinear cases.

Figure 4.1 shows again the potential extrema proportion. All vertices in the global hull (regions 4, 7, 8, 9, 10 and 12) are vertices in the local hulls, and therefore CCS extrema, since no subset of the global hull can exceed its limits. Regions 1, 6 (0+0i) and 2 (4+2i) are extrema on their local hull, regions 5, 8

and 11 are on the same edge of their local hulls, and again region 3 (5-1i) is on the hull edge. Thus from 12 regions 8 are extrema and 4 are collinear with an edge, which may or not be extrema depending on the tolerance factor used, compared to 6 in the global hull.

4.2.3 Vector Area Morphology Sieve (VAMS)

The VAMS method [143] involves getting the vector differential field of the regions. The methods begin with the reduced ordering definitions of extrema and median values; if

$$\sum_{i \in n} ||x_m - x_i||_p <= \sum_{i \in n} ||x_j - x_i||_p <= \sum_{i \in n} ||x_o - x_i||_p, \forall j \in n \tag{4.2}$$

where x_m and x_o are the median and the prime outlier from n points, taking the middle part gives a measure of extremeness $d[x_j] = \sum_{i \in n} ||x_j - x_i||_p$ from point x_j to its n neighbours.

The combined distances of all the pixels to their neighbours are averaged region-wise by the overall area, in order to assign a single value and compensate for a higher number of contributions at larger scales. The result is a region-wise gradient field, especially visible at low scales, from which maxima and minima can be extracted. Sieving happens by openings, or merging the maxima to their closest neighbour and changing the gradient field accordingly. The extrema are divided as follows:

- Maxima, which mismatch their neighbours or correspond to small regions; includes most noise sources (especially high amplitude noise), alongside edge points and general transitions (due to high gradient changes) and regions with high perimeter/area ratios.
- Minima, which are not considered in the VAMS. Figure 4.2 shows cases where the resulting gradient field is basin-shaped: the slopes around the marked points are maxima, the point itself is the minimum. From left to right, that includes smooth greyscale extrema, areas of decreasing slope and any large regions regardless of value; in other words, regions representative of their surroundings or of a certain size.

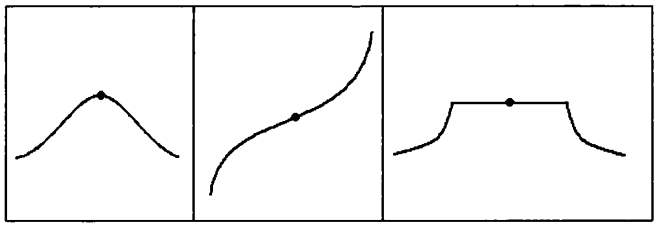


Figure 4.2: VAMS minima in 1-D greyscale

Openings sharpen the boundaries and remove noise, and will grow a region from its borders, keeping the more representative regions (minima). That means region growth depends on the location of edges and noise and therefore is image-dependent. In greyscale this method becomes the opposite of the watershed; the watershed grows a region from the minima outwards, whereas openings will start region growth from

the boundaries/ maxima inwards (which is not flooding the opposite of the gradient: points will merge to the nearest region, not grow to the nearest gradient value).

Shown below (figure 4.3) are a vector table and its VAMS point-wise and region-wise. Only two are extrema, compared to most or all for the CCS.

0+0i	4+2i	4+2i	5−1i	−3−2i
0+0i	2+4i	5−1i	5−1i	6+0i
2+4i	2+4i	5−1i	7+0i	6+0i
0+0i	2+4i	1−3i	3+0i	0+0i
0+0i	0+0i	1−3i	0+0i	−2+0i

(a) input

12	20	16	15	29
24	36	27	20	16
12	36	36	31	12
18	42	44	31	18
6	14	21	13	9

(b) pixel-wise VAMS

18.00	18.00	18.00	24.50	**29.00**
18.00	31.50	24.50	24.50	14.00
31.50	31.50	24.50	31.00	14.00
12.67	31.50	**32.50**	31.00	15.50
12.67	12.67	**32.50**	15.50	9.00

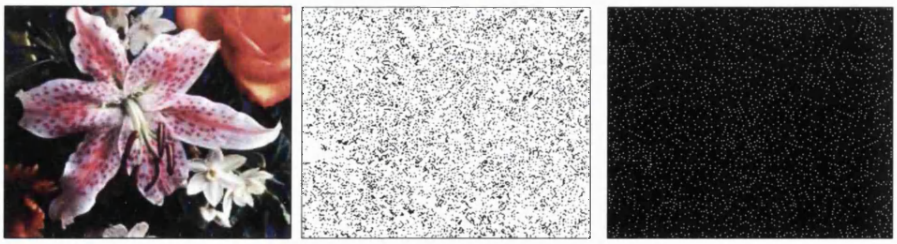
(c) full VAMS

Figure 4.3: VAMS example, 8nn connectivity: all extrema regions are in bold

4.3 Algorithmic Comparison

The CCS and VAMS differ in a number of points: their concept of extrema, the way and effects of merging regions, and their relative complexity.

The CCS method identifies extrema as points whose position on the edge of the local hull, what gets the points at the limits of the local distribution range, or the points on the outer surface, like taking the first/last rank of data in partial ordering. In contrast, the VAMS method is based on a reduced metric gradient; the metric gives a region-wise gradient in greyscale, with sharp transitions marking the outliers in the distribution. This picks the points with the least similar and uniform neighbourhood, or the noise farther from the local mean and transitions without a representative mean. Thus the CCS picks extrema position-wise and the VAMS gets those population-wise. The difference between the methods is shown in figure 4.4, with initial extrema marked in white. It is clear the CCS shows a far higher amount of extrema than the VAMS, not considering further new extrema. Also VAMS maxima concentrate at boundaries whereas the CCS has a more random pattern.



(a) input (b) CCS (c) VAMS

Figure 4.4: Initial extrema comparison

The effects of merging on the extremeness of nearby regions also differ. Both methods are difference or outlierness-based; therefore changes in the neighbourhood can induce extrema generation and removal. The CCS only takes the local neighbourhood, whereas the VAMS takes the gradient of the local neighbourhood; that means adjacent extrema are more easily produced in the CCS than in the VAMS, and extrema removal will also affect fewer regions in the local hull, since it only affects the first tier of neighbours. The CCS only allows openings, since minima are not clearly defined and probably not needed due to the speed of the method; the VAMS accept openings and closings (although the latter are better used as complements to the openings). This makes the VAMS more flexible as both can be configured independently, as seen later.

The outlier distribution clearly impacts on the sieving behaviour. The CCS will misidentify a good number of regions all throughout the images because of hull size. This proves rather aggressive, i.e. removes most minor features and not just noise or salient features, but ensures full outlier removal and an actual effect of downsampling, so this excels at segmentation and image-independent speeds. Besides, the hull is independent of space rotations and scaling hue and magnitude changes have no effect. In the VAMS, although lower extrema appear, they concentrate on edges, sharpening as much as the CCS, what goes best at noise removal. Also several output levels means ranking, so high level outliers are removed before, and the gradient field metrics allow to target specific outlier sorts (chroma/ luminance/ both). On the metrics for best fit neighbour, the original CCS method takes the best Euclidean, with YGRB channel ordering for matches; the VAMS takes scan order for Euclidean matches.

Finally, the extrema definition also affects the computation costs of the methods. The CCS needs the local hull of all points; current algorithms [146] with an order of $O(N\log(N))$ make this feasible; however, this is done for all possible extrema, so coding efficiency is key. The VAMS checks for the local gradient maxima, and then recalculates the region and only updates its neighbours; this comes down to $O(N)$, what makes it simpler.

4.3.1 Sieving and Extrema creation

As a first point in the comparison of methods there was the issue on the stability of extrema. This is developed on whether or not their numbers and locations remain stable, and if not find out if an option is open to avoid extrema creation. The section has 3 parts: showing the way extrema evolve, how they are created and how they affect performance.

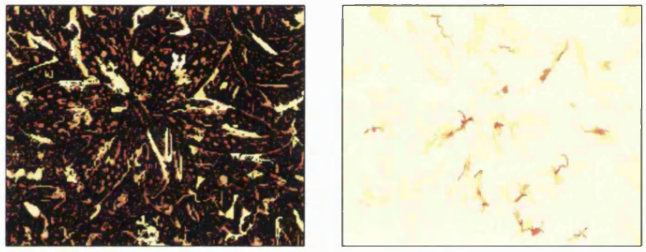
4.3.1.1 Extrema creation in normal conditions

As regions grow, maxima are created and destroyed. The sieving removes all extrema detected below threshold, not those above. This compared two parts:

- creation: the results shown in figure 4.7 are for new extrema created during each merging step as (number of regions removed from scale S - number of initial outliers at scale S); in effect, the surplus extrema produced. A positive index points to new extrema appearing up to the current scale; a negative index implies possibly extrema removal as in greyscale, with less merging than extrema yet also allowing for extrema generation and destruction at the same time.

- distribution: the results in figure 4.5 show the weights for the image regions: outliers at scale S get their weight increased by S, at all scales. Higher weights thus correspond to higher scale repeated outliers, but without a specific order. Figure 4.6 gives some of the actual distributions, with the colours inverted for the VAMS.

Graphs and images are shown for the 'lily' image. Extrema are noted in 'hot' colourmap, black showing absence and white showing the most frequent extrema; in the second example, extrema are marked in black for the VAMS, and in white and below it for the CCS. Graphs are clipped at scales 6 and above, since below that the CCS will destroy a high number of extrema by merging adjacent ones; as a note, the CCS begins with 41027 regions and 33606 outliers, merged to 16406 regions and 12874 extrema at scale 2. This rarely happens in the VAMS. The results show there is a higher extrema creation in the VAMS than in the CCS, which at points clearly shows extrema destruction. Also the distribution differs: the VAMS outliers remain stable at scale X, with the original ones or their neighbours becoming extreme at $X+1$, keeping the merging at certain spots; the CCS takes most regions at any scale as outliers, causing generalised merging.



(a) VAMS outliers (b) CCS outliers

Figure 4.5: Extrema distribution: scales 1 to 50

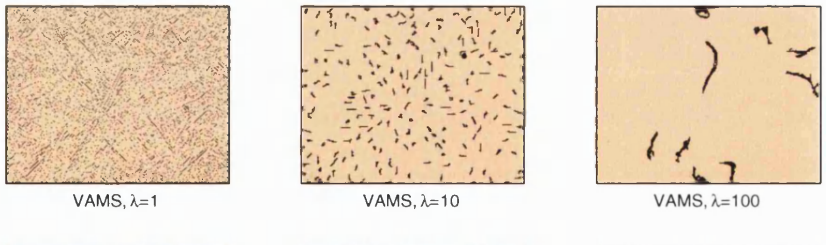


Figure 4.6: Extrema distribution: scales 1,10,100

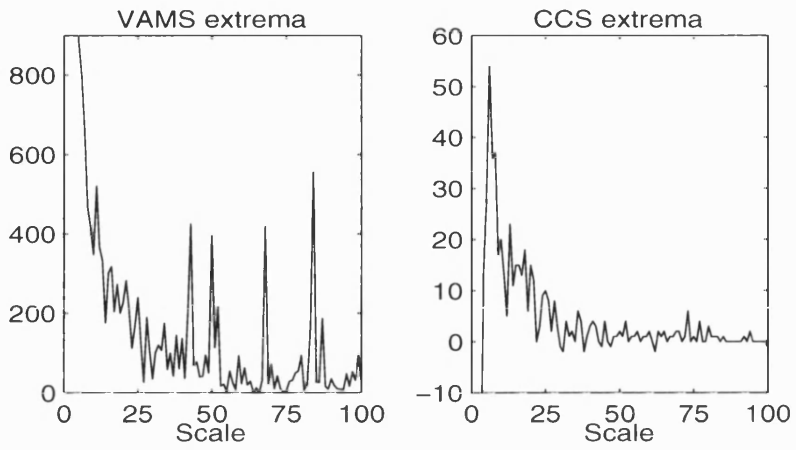


Figure 4.7: Extrema generation

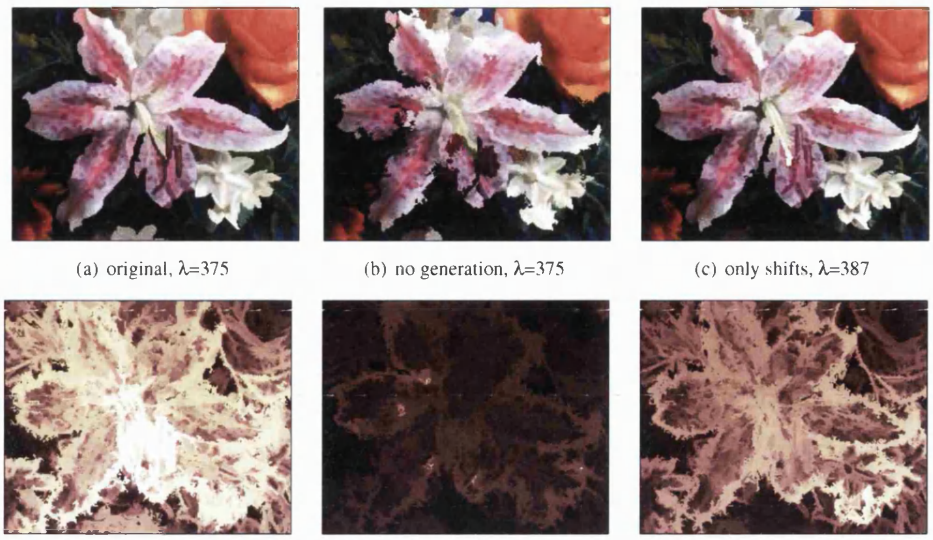
4.3.1.2 Preventing extrema creation

Since extrema are created by default, there is also a chance of certain merging sequence which prevents its creation at any scale, as in greyscale. Since the CCS takes most points as extrema and all extrema are considered equal, this was not evaluated as the creation of extrema proved different. The test was carried out on the VAMS method with 2 options:

- forbidding the creation of new maxima: like in greyscale, maxima can only expand or disappear when adjacent to other maxima.
- allowing maxima shifts: this considers the fact that the VAMS metric generally decreases with region size. Only one of the original neighbours of a destroyed maximum can become a maximum, and maxima may be destroyed if other same-level maxima are kept, hence the shift. This is more tolerant, and more adequate to the presence of adjacent maxima in both methods.

Merging was done to the best-fitting neighbour which complied with the rules above. Both tests failed to achieve full segmentation; hence this can be considered an empirical proof for the limitation in this method. Results are shown for both variants using the L_2 metric.

It can be seen in figures 4.8b-c that extrema shifts allow for further progress and less distortion, but the result is the same: the presence of equal-valued points will induce multiple new extrema, eventually barring all local maxima seen in figures 4.8e-f from merging, which when using extrema shifts translates into stopping after a few merges with a choice of values rather than a long chain of dissimilar ones, explaining its lesser simplification and the clearly worse merge choices of not taking shifts compared to the original VAMS sieve. The CCS was not evaluated; this is because of high extrema numbers, which makes extrema over-creation far less likely than removal, and no measure on extremeness shifts; however, this problem would probably be present at some point, especially when points on the global hull must be removed before those inside the local hulls. A different point of view is that if a region falls on its own hull and its neighbours, then it can induce more extrema unless the new value is at a higher value than the



(a) original, λ=375 (b) no generation, λ=375 (c) only shifts, λ=387

(d) VAMS gradient field (e) gradient, no generation (f) gradient, only shifts

Figure 4.8: Extrema generation test

points inside the hull, what can be simply solved by merging to minima or by ensuring no minima exist (note the latter is easier for 3-D spaces than for 2-D).

All of this points to the inability to avoid new extrema in these sieves, at least when its definition is neighbour-dependent and an exhaustive search on all merging sequences is not available. Hence, step 3 of the algorithm in section 4.2.1 is a requisite for idempotency.

4.4 Colour Sieve Evaluation

4.4.1 Timings

Results follow for the 'lily' image (shown before) in seconds. 8-connectivity and the Euclidean metric were used in all cases. The implementation for colour methods used follows a priority-queue structure with optimised extrema listing; for the greyscale methods, these are based on the same structure to prevent any bias, and extrema are identified by their luminance (R+G+B). For the colour-based sieve (GS-AO, GS-AOC) used throughout the thesis, the best neighbour is selected based on the metric distance, not on the nearest greyscale value (Y-AO, Y-AOC) as in chapter 3. The former prevents the artifacts of conditional ordering where equal-luminance regions are simply merged, yet may induce extrema shifting (see previous section).

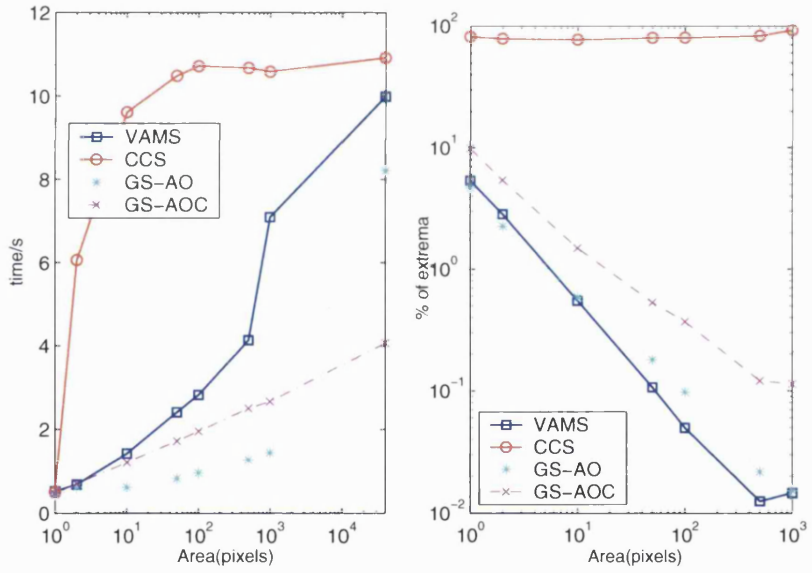
The Tarjan's union method is that implemented by Nick Young[147]. Its alternating filter (ASF) is found by taking Tarjan's method at alternating increasing scales: the original algorithm [119] does not offer an explicit AOC sieve.

It is clear that the key factor in timings is the number of new extrema present at any scale. The VAMS

Table 4.1: Timings for area morphology, colour and greyscale

scale	1	2	10	50	100	500	1000	40000
Colour								
VAMS	0.521	0.681	1.422	2.413	2.825	4.135	7.091	9.994
CCS	0.461	4.937	7.15	7.28	7.610	7.682	7.591	7.681
GS-AO	0.431	0.621	0.620	0.832	0.961	1.272	1.442	8.212
GS-AOC	0.460	0.701	1.212	1.722	1.953	2.504	2.664	4.066
Greyscale								
VAMS	0.521	0.641	1.332	2.203	2.524	4.206	5.908	12.498
CCS	0.371	0.560	1.072	1.243	3.225	8.532	11.787	26.608
Y-AO	0.391	0.421	0.550	0.802	0.951	1.572	1.793	75.278
Y-AOC	0.430	0.591	0.922	1.351	1.633	2.503	2.965	17.345
Tarjan AO	0.019	0.019	0.020	0.019	0.019	0.019	0.020	0.022
Tarjan ASF	0.030	0.059	0.289	1.472	2.951	14.741	29.422	1108
Queue AO	0.002	0.029	0.054	0.118	0.195	0.411	0.560	44.484
Queue ASF	0.003	0.033	0.801	20.850	84.246	3618	$2.6 * 10^4$	10^7_*

* Extrapolated



(a) Timings (b) extrema

Figure 4.9: Performance with scale

has similar number of extrema, and behaviour, as the greyscale methods, with a few representative outliers evenly surrounded by non-extreme pixels which gradually expand and decrease rapidly in numbers.

In contrast, the misclassification of small (4 point) hulls in the CCS gives a high, stable number

Table 4.2: Extrema / regions for colour sieves

scale	1	2	10	50	100	500	1000
VAMS	2209	1054	151	21	8	1	1
	41207	37199	27547	19660	16102	8024	6871
CCS	33616	12852	2211	456	223	44	23
	41207	16402	2878	572	279	53	25
GS-AO	1926	879	207	55	28	5	3
	41207	38950	35119	30554	28631	23074	19723
GS-AOC	4062	1973	424	110	63	11	7
	41207	36553	28457	20697	17040	9114	6150
Y-ASF	4062	2042	539	164	104	37	21
	41207	36845	30116	23548	20739	13029	9888

of extrema which effectively produces an effect of sub-sampling as the number of regions is roughly inversely proportional to scale and partly overcomes the much higher initial cost of this method, proving to be the fastest when no methods are optimised. Although this proves a benefit in applications where all regions must achieve a certain size, it can potentially induce more distortion than desired for purposes other than segmentation, as in noise removal in the next section.

In greyscale, Tarjan's method surpassed the area opening implemented, and the more classic max-tree method with a constant time performance. That was not the case with the AOC, which meant multiple Tarjan sieves. This reflects the main problem of colour sieving and alternating filters: unlike in the greyscale opening, new extrema appear/disappear and updated vector values are required, which means updating the image and outlier lists at each merge rather than just at the end.

That is the weakness of the slower methods, where slowly growing regions need more updates and pixel sorting for $O(N^2)$ whereas the best case goes at $O(N\log(N))$; a different implementation based on region rather than pixel comparison would shift costs from pixel operations to neighbour lists. Here is the proof for the timings:

The cost of merging 2 sorted pixel lists R_a and R_b (order N) and of finding the new neighbours by removing the set intersection between M neighbours and N pixels (order $M+N$) with $M = \alpha N$, connectivity *conn* and $0 < \alpha < conn$ gives a cost C

$$C(a \bigcap b) = (R_a + R_b) + (R_a + R_b + M_a + M_b) \tag{4.3}$$

In the worst case only 1 region grows up to N ($N \gg 1$). That means $M_b \simeq \frac{conn}{2}$ except at borders. M_a is less predictable, always peaking halfway through. For a raster scan merge $M_a = \sqrt{N}$ and $M_b = \frac{conn}{2}$, except for the first and last $\sqrt{N}$ merges.

The worst possible case (a 4-connected spiral or an 8-connected mesh) gives $M_a \simeq R_a$ and $M_b = conn$, decreasing after $M_a > \frac{N}{2}$. Assuming M_a decreases linearly and $\sum f(x) \approx \int f(x)dx$ for $N \gg 1$,

$$C(I) = \sum_{x=1}^{N-1}(x+1) + (x+1+M_a+conn) \tag{4.4}$$

$$C(I) \simeq N^2 + (2 + conn)N + \int_0^{N/2} x dx + \int_{N/2}^{N} (N - x) dx \qquad \text{if } N \gg 1 \tag{4.5}$$

$$C(I) \simeq N^2 + (2 + conn)N + \frac{N^2}{8} - 0 + \frac{N^2}{2} - \frac{3N^2}{8} \tag{4.6}$$

$$C(I) \simeq \frac{5N^2}{4} + N(2 + conn) \tag{4.7}$$

$$O(C(I)) = N^2 + N$$

In the best case half of the R regions are merged to same A area regions (A=N/R) at each step. Also $M_a = M_b \simeq 4\sqrt{A}$ for the most square, compact regions.

$$C(I) = \sum_{x=1}^{\log_2 N} \frac{R}{2}((A + A) + (A + A + 2 * 4\sqrt{A})), A = 2^x \tag{4.8}$$

$$C(I) = \sum_{x=1}^{\log_2 N} R(\frac{N}{R} + \frac{N}{R} + 4\sqrt{2^x}) \tag{4.9}$$

$$C(I) = \sum_{x=1}^{\log_2 N} (2N + 4 * 2^{x/2}) \qquad \text{, the sum of a constant and a geometric series} \tag{4.10}$$

$$C(I) = 2Nlog_2N + 4\sqrt{2}\frac{1 - 2^{0.5log_2N}}{1 - \sqrt{2}} \tag{4.11}$$

$$C(I) = 2Nlog_2N - 4\sqrt{2}(1 - 2^{log_2\sqrt{N}})(1 + \sqrt{2}) \tag{4.12}$$

$$C(I) = 2Nlog_2N + 4\sqrt{2}(\sqrt{N} - 1)(\sqrt{2} + 1) \tag{4.13}$$

$$O(C(I)) = N \log N + \sqrt{N}$$

That ignores elements as priority queues ($O(N \log n)$) and extrema finding, giving an order closer to $N^{3/2} \ldots N^2$. For 'lily' (186*230), the times for the 8-connected sieves to image size for colour VAMS, colour CCS and GS-AO on the luminance are 10.0, 10.9 and 75.6 sec; for 'lenna' (512*512), the times are 132.0, 140.9 and 1861.6; using 4-connectivity halves VAMS timings, lowers the CCS results by a third and has little impact on greyscale. Further tests confirmed this, giving a $N^{3/2} \ldots N^2$ link for increasing scales

4.4.2 Noise performance

The data shown are for the average of the same set of figures as for median evaluation: 'lily', 'boats', 'autumn', 'lenna', 'baboon' and 'sample1'. Results are scales by 10^{-2} and use the L_2 metric and 8-connectivity. All methods were sieved to the scale (from area 2 to 150) that gave the minimum average NMSE results for each sieve and noise model, or to area 10 for the original images.

The noise sources were: Gaussian at $\sigma^2 = 10^3$, impulsive at $\rho\% = 10$ and a combination of both, either with independent (ε=0%) or 50% correlated noise among channels. Note that the distortion generated by the methods is low enough to have an effect when using lower noise settings, e.g. Gaussian with $\sigma^2 = 100$, although this is more visible in the settings used, therefore these are not shown for simplicity.

Filter	input	Gaussian		Impulsive		Mixed
		$\varepsilon = 0\%$	$\varepsilon = 50\%$	$\varepsilon = 0\%$	$\varepsilon = 50\%$	$\varepsilon = 0\%$
none	0	4.1323	4.1081	10.3597	6.8632	14.0668
VAMS	0.5133	2.1678	2.1563	0.8448	0.4951	2.7556
CCS	0.7817	2.1111	2.2104	1.0017	0.6510	2.7160
GS-AOC	0.3987	2.6374	2.0134	1.1952	0.6424	3.6915
Y-ASF	0.3028	1.2316	1.2244	0.4261	0.2909	1.4993
VMF	1.2032	2.0617	1.9394	0.6965	0.5926	2.3180

Table 4.3a: NMSE vs method to best scale

Filter	input	Gaussian		Impulsive		Mixed
		$\varepsilon = 0\%$	$\varepsilon = 50\%$	$\varepsilon = 0\%$	$\varepsilon = 50\%$	$\varepsilon = 0\%$
none	0	15.1076	12.7246	9.4276	4.7711	21.2566
VAMS	1.3575	9.2344	8.5659	1.6071	1.1213	10.2343
CCS	2.2627	8.4017	8.0174	2.0847	1.8133	9.0298
GS-AOC	1.0858	12.1008	9.0331	2.1215	1.2376	12.0578
Y-ASF	1.1468	6.1376	5.3458	1.7182	1.1806	6.7083
VMF	2.3646	7.8162	6.8559	1.9335	1.7737	8.6781

Table 4.3b: MCRE vs method to best scale

(a) corrupted image (b) marginal ordering (c) luminance ordering

(d) TVMF, mask 11*11 (e) VAMS sieve (f) CCS sieve

Figure 4.10: Colour sieving on 'boats' image, to scale 100

Results show both methods are strong against impulsive noise, and their performance is better than for the adapted greyscale sieve (GS-AOC), and comparable to the median methods (see prior chapters); medians work best at a 3*3-5*5 scale. However, they are still worse than marginal filtering (Y-ASF), where the presence of new colours removes most of the noise. Both methods are also similar at this scale, with the VAMS being slightly better against Gaussian noise in terms of MCRE. However, at this scale the aggressiveness of the CCS method still is an advantage, making it better for Gaussian noise

removal by removing most of the samples. The CCS is best against Gaussian: most points are extrema, so the median-like effect helps to smooth the image, whereas the VAMS will be restricted after removing the main patches until large scales. For impulsive, the VAMS wins by just removing the noise, which effectively reaches a minimum at scales 5-10, whereas the CCS goes on merging. Mixed Gaussian and impulsive noise looks like high-variance noise, and filters behave accordingly. In terms of appearance, it is clear in figure 4.10 the new colour sieves are an improvement over conditional ordering and give less edge blurring and rounding than the truncated median at a similar scale; however, the presence of random colour patches at the most corrupted areas, where the presence of new hues in marginal morphology is an advantage, still makes them inferior to the latter.

Finally, something not included on the results is the fact that Gaussian noise improves the timings for the VAMS depending on variance, but not for the CCS, most likely due to the gradient, and hence noise, dependency of the VAMS and to the already high number of seeding points for the CCS. This tunes the amount of sieving in the VAMS, making it less aggressive in noiseless images. To summarize and as seen in figure 4.10 and tables 4.3a-4.3b, slight touch-ups and impulse removal are the strong points of the VAMS, whereas high scale fast segmentation and Gaussian removal are the strengths of the CCS and both prove better than conditional ordering, although if lack of causality or new hues are not important there are better denoising techniques (i.e. medians, marginal sieves).

Chapter 5

New Colour Sieves

The methods examined before are a improvement over usual colour morphology. However, they prove less than ideal, and given they are recent (2003) and the lack of other colour sieves, except maybe the datasieve by Iyer [148], there is therefore place for developments. The methods developed all follow the properties of the original colour sieves.

5.1 VAMS developments

5.1.1 VAMOCS - VAMS M-sieving

Both maxima and minima in the VAMS scalar image can correspond to outliers; maxima generally correspond to noise spikes and sharp transitions whereas minima include any region too large and compact to be an outlier, hence its neighbours are maxima, and features with gradual transitions as in figure 4.2, forming a gradient basin. Since the original method relied only on openings, a variant was developed using opening-closings, to remove all extrema, both noise and features. This was aimed at boosting its effect by increasing the number of seeds, especially on uniform regions where image quality would not change significantly, and equally to speed up segmentation, a side effect of higher seed numbers.

After initial trials, it was found treating maxima and minima the same (i.e. removing them) was not adequate; this was solved by using merging constraints on the minima, following on Salembier's rules [56]. This reduces to swapping the region order when merging, so that minima can stand merging if not considered as outliers. Some constraints were tried to remove the less relevant minima, those too compact or different, while keeping all others. The best ones turn out to be area, together with extremeness per unit surface and area. That means all minima larger than their neighbours or with a lower VAMS field score per neighbour are kept rather than removed.

The differences show up in figure 5.1. Openings remove noise and small features and sharpen boundaries, but preserve too many small and uniform features surrounded by large uniform zones (e.g. most features in the rose, the white flowers at scale 1000) which are not relevant at that scale and similar in colour. Closings alone grow large uniform regions yet ignore maxima (edges and noise) like a watershed

flooding, leaving a series of flat zones at higher scales (area 1000,10000); unlike maxima, few if any minima appear or 'shift' as described in section 4.3, which explains why some non-maxima features remain even at scale 10000 (e.g. the white flowers, the lily's stamen) and the observed worst-case timings shown in equations 4.4-4.7. In contrast, the VAMOCS looks like the VAMS openings, although without the artifact of small, isolated minima at the same scales in the VAMS and many less regions at each scale, making it the best option in speed and segmentation by removing both noise and nearly flat zones.

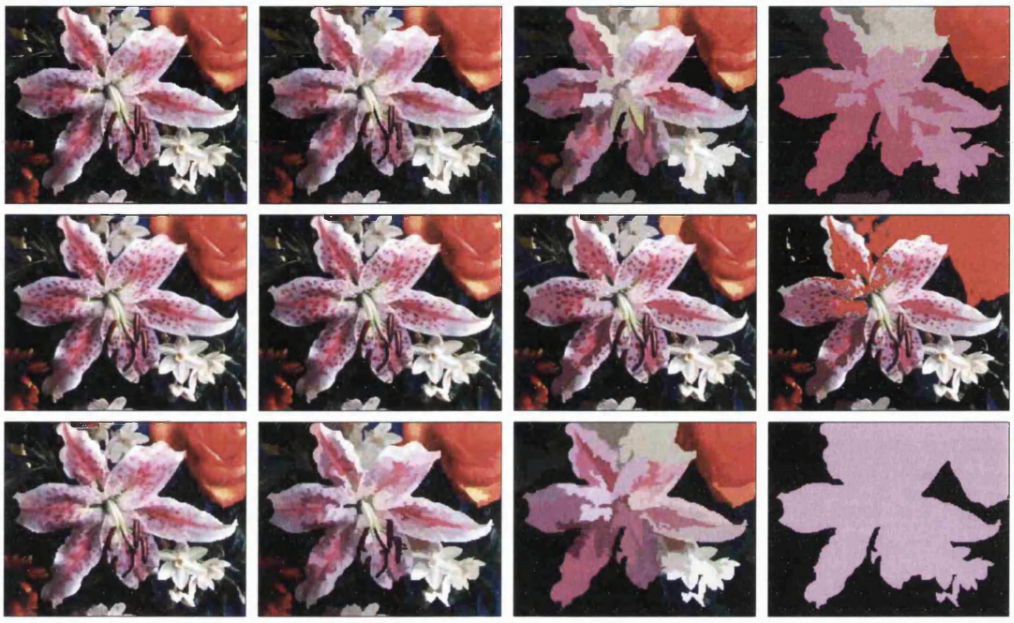
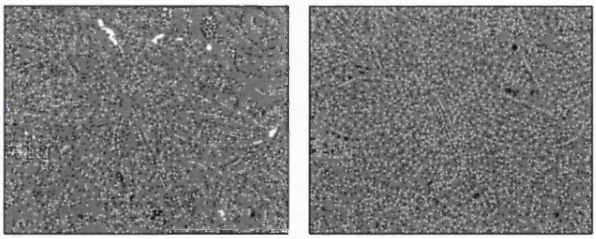


Figure 5.1: VAMS family sieve comparison. Top to bottom: VAMS openings, closings and VAMOCS. Left to right: area = 10, 100, 1000 and 10000.



(a) GSAOCS (b) VAMOCS

Figure 5.2: Initial extrema comparison: maxima in white, minima in black

From this last point and since openings and closings are possible, different image-based merging conditions are possible on both of them, and the distance-based gradient, though space-dependent, allows customisation in terms of rules and merging decisions. For example, colour-oriented metrics aimed at noise removal can be replaced by more robust yet less hue coherent ones for segmentation, and maxima

and minima can be kept or removed in terms of region metrics.

5.1.2 VDMS - VAMS distance normalisation

The VAMS extremeness metric is formed by taking the metric differences for all the pixels in the region, and then averaging by their total area to counter the increasing contributions, and extremeness, with scale. However, the number of distances considered, and thus extremeness, is proportional to the region's perimeter, not to its area. This means large regions and those with fewer neighbours are more prone to becoming minima, effectively allowing regions with a low surface/area ratio or simpler shapes to survive merging regardless of scale. Since this is the main reason why extrema creation in the VAMS is unavoidable, another variant using perimeter or surface distance, rather than area, to counter metric growth is proposed, giving the Vector Distance Morphology Sieve (VDMS).

0+0i	4+2i	4+2i	5−1i	−3−2i
0+0i	2+4i	5−1i	5−1i	6+0i
2+4i	2+4i	5−1i	7+0i	6+0i
0+0i	2+4i	1−3i	3+0i	0+0i
0+0i	0+0i	1−3i	0+0i	−2+0i

(a)

18.00	**18.00**	**18.00**	24.50	**29.00**
18.00	31.50	24.50	24.50	**14.00**
31.50	31.50	24.50	31.00	**14.00**
12.67	31.50	**32.50**	31.00	15.50
12.67	**12.67**	**32.50**	15.50	**9.00**

(b) VAMOCS

6.00	**4.50**	**4.50**	5.16	**9.67**
6.00	**6.63**	5.16	5.16	**3.50**
6.63	**6.63**	5.16	3.88	**3.50**
5.43	**6.63**	5.91	3.88	3.88
5.43	**5.43**	5.91	3.88	**3.00**

(c) VDMOCS

Figure 5.3: Extrema for VAMOCS and VDMOCS: maxima and minima in bold

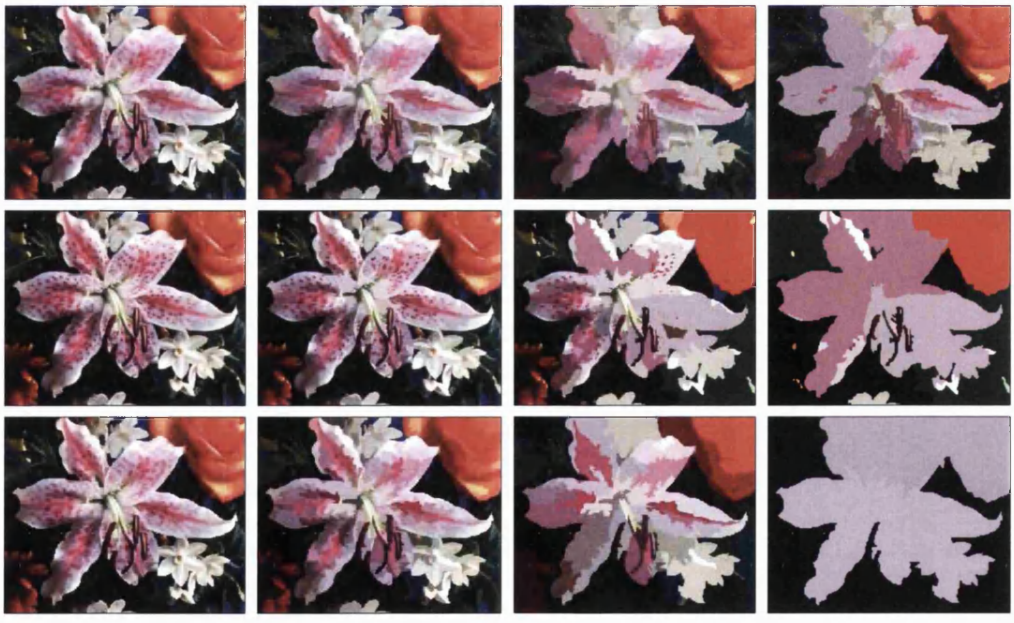
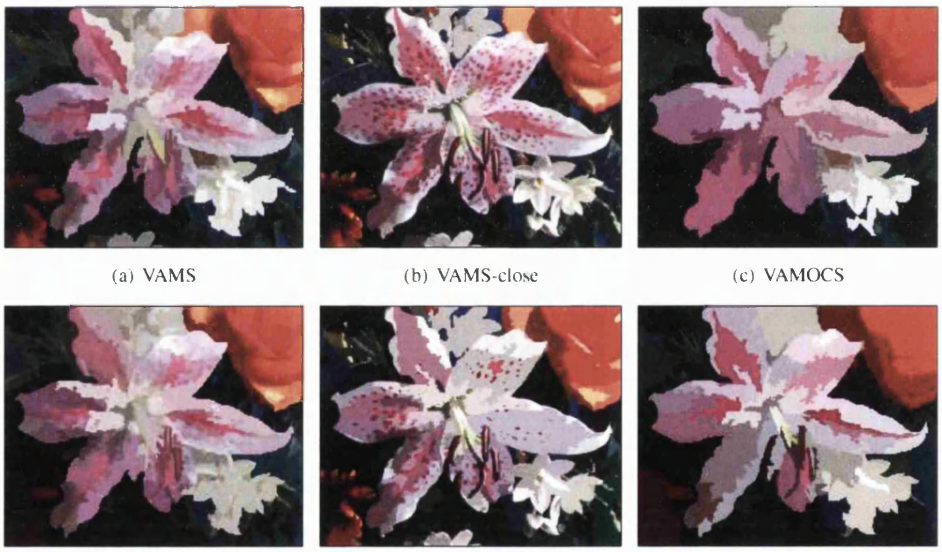


Figure 5.4: VDMS family sieve comparison. Top to bottom: VDMS, closings and VDMOCS. Left to right: area = 10, 100, 1000 and 10000.

As seen in figure 5.4, the VDMS behaves quite differently. The new variant has a region-wise scalar image less biased against region size after the initial scales (where noise is still removed), reducing maxima generation while boosting that of minima, with outliers mapping to maxima regardless of scale or complexity unlike in the first row of figure 5.3, where simplification is uniform as opposed to the few, large regions enveloped by smaller regions seen at scale 10000 for the VDMS, resembling the VAMS closings. In contrast, the VDMS closings in the second row of figure 5.4 give far more simplification than either VAMS closings or openings (other than keeping small maxima regions e.g. the lily's stamen and texture). Results for the VDMS thus prove visually similar up to moderate scales, since the better identification of extrema at all scales is countered by the increased extrema generation and the enhancement of simple-shaped, regular regions in the original method. This makes this variant more correct, yet less effective, at fast segmentation.

This variant can also be extended to include closings, giving the VDMOCS. This, however, includes a minor change. The VAMOCS reverses the merging order for all minima, except for those smaller and with more extremeness per unit neighbour (VDF/Area/Neighbours) than its surroundings; the VDMOCS takes instead extremeness per unit area (VDF/Perimeter/Area) together with area since the shape dependency is absent. That slightly affects the measure for complex regions.



(a) VAMS (b) VAMS-close (c) VAMOCS

(d) VDMS (e) VDMS-close (f) VDMOCS

Figure 5.5: Difference in extrema split, area 1000

Another difference is the split of extrema between openings and closings. In the VAMS, new maxima are generated as regions grow (lowering their extremeness per unit area) and become local minima; that gives an uniform granule distribution over scales, i.e. no fixed seeds. For closings the opposite happens; minima achieve even lower values while merging and increasing their area, resembling greyscale closings in behaviour and timings. In the VDMS the opposite case happens: maxima retain their identity for a range of scales, being independent of region size, whereas minima are frequently created and destroyed

and timings are equally reversed. Figure 5.5 shows this, and more scales are in figures 5.1 and 5.4. VAMS closings are clearly less aggressive than the VDMS ones which, over the same range, affect far more regions in a less consistent manner; the situation is almost reversed for both types of openings, which is clearer at a scale of 10000 in figures 5.1 and 5.4 where the VDMS removes most of the regions. Their combinations (fig. 5.5c, 5.5f) behave and look alike, with the VDMOCS keeping some small and complex features (e.g. the lily's stamen, some highlights) whereas the VAMOCS is biased towards more compact regions.

5.2 PCS - Positional Colour Sieve

The two original sieves have different strengths and weaknesses: the VAMS has a continuous range of extrema classification, is not too dependent on the connectivity and number of adjacent regions and detects points not representative of their population, but has a low extrema count. The CCS, on the other side, is scaling and rotation invariant, detects positional extrema regardless of the distribution of all other points and can find all spatial outliers, but proves too aggressive and reliant on the number of adjacent regions compared to that of channels.

The Positional Colour Sieve (PCS) attempts to combine their strengths in terms of outlier detection, according to position and distance [149], while being similar to an ASF for greyscale images. This method takes features from the 2 methods studied: the hull and the VAMS. The number of points above/below a plane through the point is counted and normalised, and the value considered a measure of outlierness, giving a vector differential field as in the VAMS with minima at the transitions. The method adopts the M-N sieve behaviour when in greyscale images, and performs like the convex hull in the CCS. Similarly to the convex hull, spatial outliers are often picked out; like the VAMS, a multilevel gradient image means that only the local maximum will be removed, with less adjacent extrema while those akin to their neighbours get lower scores. Two variants were used: the first, marginal one considers the outlierness for each of the image components, combining them with a L_2 metric; the second takes the plane normal to the vector between the region and the mean centroid of its neighbours for the measure of outlierness.

The process for either version is:

1. For each axis, count the number of neighbours (not adjacent regions) above and below the region and subtract them. If all points are on par and above/below, add the number of equal values to avoid an undersized maximum; this does not happen for the marginal PCS.

2. Normalize by the number of dimensions then take the magnitude of the resulting vector (i.e. Euclidean of the marginal differences).

3. Normalize by the number of neighbours to avoid increasing numbers with surface. The limit is thus 1; equal-level points are ignored and assumed to lie equally distributed to each side.

Shown in figure 5.6 are a vector field, its marginal PCS metric field, and the hull for region $(5 - i)$ in the PCS with the degree of extremeness as seen at any individual point within, blue indicating the lowest values and red the highest. The PCS has 4 maxima, compared to 3 maxima and 4 minima in the

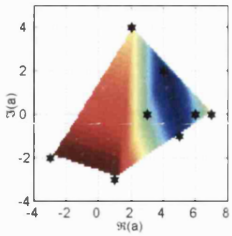
GS-AOC, 1 maximum and 4 minima in the L_2 Y-AOC, 2 maxima and 4 minima for the VAMOCS and 8 to 12 maxima for the CCS (see last chapter), making it more selective in that aspect; in addition, these outliers reflect points with an extreme position in one or either component, hence finding the positional extrema as desired.

0+0i	4+2i	4+2i	5−1i	−3−2i
0+0i	2+4i	5−1i	5−1i	6+0i
2+4i	2+4i	5−1i	7+0i	6+0i
0+0i	2+4i	1−3i	3+0i	0+0i
0+0i	0+0i	1−3i	0+0i	−2+0i

(a) input

1.0000	0.4714	0.4714	0.5530	**1.0000**
1.0000	0.7329	0.5530	0.5530	0.5893
0.7329	0.7329	0.5530	**0.7906**	0.5893
0.7211	0.7329	0.7693	0.3187	0.4518
0.7211	0.7211	0.7693	0.4518	**0.7071**

(b) PCS



(c) PCS score on hull

Figure 5.6: PCS field

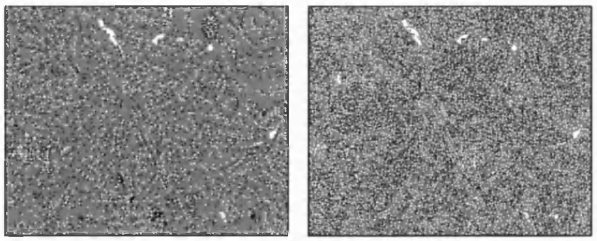
Example: Let the input data be {(0,0) (1,2) (3,1) (5,2) (7,6)} with the (5,2) region 4-connected to all other regions and all regions having area 1.

- For the marginal PCS, (5,2) has 1 point above, 3 below in the first dimension and 1 above, 2 below in the second dimension. That gives totals of -2 (1-3) and -1 (1-2); the magnitude is $\sqrt{(2^2+1^2)/2} = 1.58$, which normalised by 4 neighbours gives a metric of 0.39.
- For the vector PCS, the mean of the neighbours is (2.75,2.25). The vectors from this mean are {(-2.75,-2.25) (-1.75,-0.25) (0.25,-1.25) (2.25,-0.25) (4.25,3.75)}. The dot product from (2.25,-0.25) to itself is 5.125, hence dot products greater or less than this are on different sides of the 'boundary' across (5,2) and perpendicular to the axis passing through the mean and (5,2). The values for the 4 neighbours are {(-5.625) (-3.875) (0.875) (8.625)}, giving 1 above, 3 below and a total of -2; the metric is $\sqrt{(-2)^2)/1}/4$ or 0.50.

The extrema detected again can be maxima or minima:

- maxima are points on or near the hull, especially those which are more different from the neighbours (i.e. clear or peak outliers). For the marginal variant, points on the hull are ignored if others are more extreme, i.e. transitions are not maxima.
- minima are points near the centroid and within the hull (i.e. smooth transitions)

Closings will just sharpen region boundaries; openings merge regions on a M/N sieve fashion, ignoring the edge pixels. Their combination works like that for the VAMOCS, see figures 5.2 and 5.7. Merging is similar to that in the convex hull method, being scale invariant and also rotation invariant for the second version, but customisation is not available: centroids and hull scaling improve some issues but are essentially fixed, and tolerance to transitions can be increased by taking the position in the hull at the expense of missing outliers.



(a) GSAOCS (b) PCS

Figure 5.7: Initial extrema comparison: maxima in white, minima in black

5.2.1 Vectorial PCS

The method was further developed, combining it with PCA to pick the axes with the largest variance. That would expectedly find extremeness within the local hull, not the colour space, by aligning it against the general trend. This improved the results, although exposed the inherent problems of the initial method, the axes. Using simply the PCA axes gave a poor performance when faced with impulsive noise, since none of the axes for the overall neighbourhood is aligned, and thus responsive, to it. Adding more weight the central pixel, from $N = 1$ to $N = 50\%$ gave adequate noise rejection; however, this was also achieved by arranging one of the axes to be aligned with the mean to central pixel line, giving a simpler method. For clarity, this version is called PCS-V (vectorial).

5.2.2 Angular PCS

The PCS was again redefined but with the same logic behind, i.e. the removal of points according to their position, reducing to a classic sieve for greyscale images. While regarding points on the hull more extreme than those inside and distant points more extreme than nearby ones, the measure was changed: rather than using fixed axes or PCA to define local axes (an alternative with moderate success), the angle and distance were selected. This is referred as PCS-A.

This version considers:

1. The angle with neighbouring points. For points on the exterior of the hull the angle is measured using neighbouring exterior points. For points inside the hull the maximum angle to neighbouring exterior points is used. This reduces to a boolean test in greyscale, and using the wedge product [1] or other vector operators it can be extended to higher dimensions. When all points are in a plane the maximum angle is π so, to avoid bias, all angles are scaled to a $0 - \pi$ range. As the angle reduces the point becomes more extreme, so it is subtracted from the maximum angle to give an increasing measure.

2. The normalised distance, measuring the extremeness of the position. The first step in finding this measure is to define 3 points, the original point (o), the point furthest from o (f_1) and the point furthest from f_1 (f_2). The normalised distance is then given by $(\overline{of_1} + \overline{of_2})/\overline{f_1f_2}$ or $2\overline{of_1}/\overline{f_1f_2} = 2$ when o and f_2 are the same point. The measure is shifted to a range of 0 to 1.

A measure of extremeness is then given by the product of the angle and the normalised distances measures. This measure overcomes the problem of using the angular measure in isolation, as points that are close to coplanar can be differentiated by the distance measure. Similarly, points that are located amid similar neighbours are less likely to be classified as extreme. Finally, a third criterion is added to ensure that points within the hull are not classified as extreme:

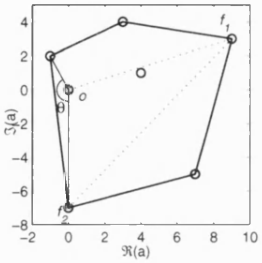
3. Whether the point is within or on the edge of the hull. As extrema are not expected to occur within the hull, the scalar product of measures 1 and 2 is augmented by π for points on the edge of the hull. Points coplanar or collinear with edges are considered to be in the hull.

0+0i	3+4i	3+4i	2−3i	4−5i
4+1i	4+1i	3+4i	0−7i	3+2i
9+3i	0+0i	0+0i	−1+2i	3+2i
9+3i	7−5i	7−5i	−1+2i	2+1i
0+2i	0+2i	7−5i	−1+2i	2+1i

(a) Complex image

5.6009	4.8543	4.8543	0.0040	3.4123
3.1419	3.1419	4.8543	**5.0654**	4.5705
5.0680	0.1045	0.1045	4.3594	4.5705
5.0680	5.0877	5.0877	4.3594	3.3012
5.3871	**5.3871**	5.0877	4.3594	3.3012

(b) PCS-A scalar image



(c) convex hull for $0 + 0i$ region

Figure 5.8: PCS-A sieve example using 8nn connectivity, example measurement.

In this case and since 3 channels were used, the angle definition was the solid angle, or the area supported by the points projected onto a unit radius sphere. This was taken using the 3 most distant points; a more exact value would be obtained by taking all points within sight, but this was not implemented due to complexity and timings. The relative distance used (ratio of distances to farthest point and farthest from it to distance among them) is used to prevent local consent (a broad angle) from masking a true outlier and to remove some unclear outliers; this proved better than using the angle among all three points, its cosine or the unclipped ratio, and behaves properly when in greyscale. In addition, hull membership is included as a factor; hull extrema (not including coplanar points) have higher values than the rest.

The result is a sieve that concentrates on intensity maxima in greyscale, removes most noise in colour and respects transitions while being rotation invariant. Results are superior in terms of noise but not in terms of segmentation or timing. Possible enhancements would include new distance measurements, takings the broadest angle possible with partial hull resolution, or measures of its overall shape. Also some noise escapes the action, particularly at sharp edges; the logic behind such effect is most points are a transition between the black and white points in the RGB space, or the presence of an outlier also makes nearby points more outlying. The first cause mainly applies for the angle; the second is related to the distance used.

This process is illustrated for the $(0 + 0i)$ region in the local hull in figure 5.8:

- The maximum angle to any other points is $\arccos(-14/\sqrt{5 * 49}) = 2.678$ rad (using points $(0-7i)$ and $(-1 + 2i)$), giving a measure of $\pi - 2.678 = 0.464$ rad.

- Point $f_1 = (9 + 3i)$ and is 9.49 from point o (0+0i), and point $f_2 = (0 - 7i)$ is 13.45 from f_1 and 7.00 from point o.
- Since $f_1 f_2 \neq o f_1$, points o and f_2 are not the same so there is no need to scale d_2. The normalised distance is $(9.49 + 7)/13.45 - 1 = 0.225$
- The final measure is $0.464 \times 0.225 = 0.1045$.

5.3 Evaluations

5.3.1 Timings and extrema

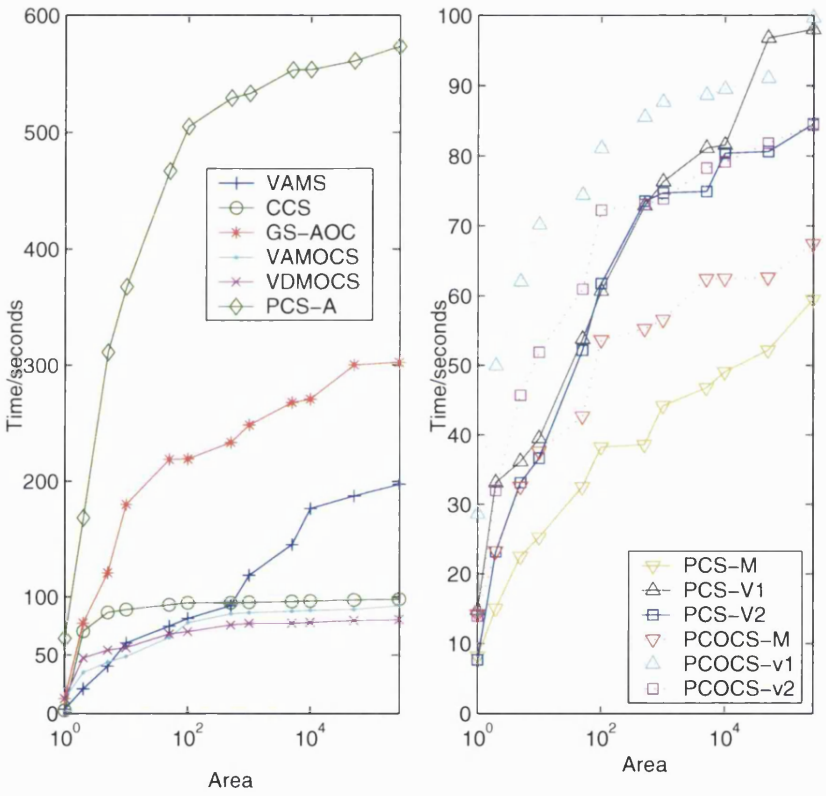


Figure 5.9: Timing for all methods

All colour methods described in the last two chapters were evaluated using 8 connectivity, Euclidean distance, and tested on the 'lenna' test image. The reasons for choosing this image over 'lily' are many. It is a more popular image in image processing, allowing for a comparison of results against other authors. It is also a larger (by a factor of 6), fairly uniform image yet with different textures (and extrema rates) at some locations, which gives a closer idea of the processing times for commonly sized pictures. Finally,

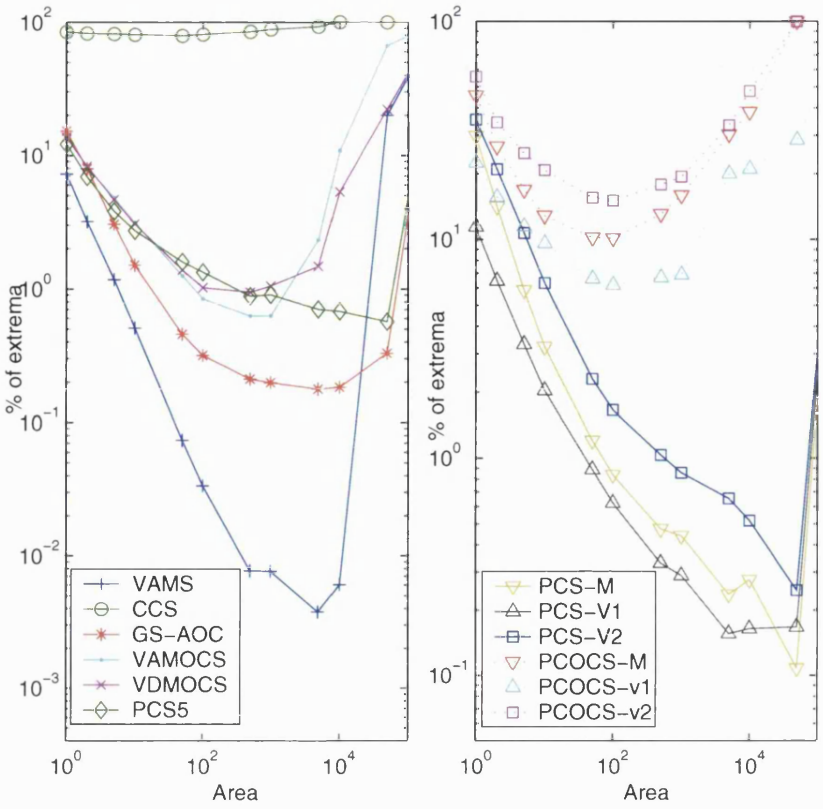


Figure 5.10: Extrema for all methods

the proportion of initial extrema is fairly similar to that of 'lily' despite differing contents, as seen in [62], so the effect on running times can be mainly sourced to image size.

Results for the CCS clearly differ to the 'lily' image; it is no longer the slowest method, being on par with most others. In the other methods this image shows the problem of the algorithm's implementation: with more key features up to high scales, those methods with a tendency to grow from a few stable seeds (VAMS, PCS-A, greyscale sieves) have greater running times than those with more random extrema.

The two variants of the VAMS M-sieve also show their similarity. The number of extrema is slightly higher for the VDMOCS (perimeter normalised), giving lower overall times but a slower beginning.

A common feature of the PCS methods is the greater number of seeds and (unlike the VAMS) increased timings for combined openings and closings, from 10% upwards; this suggests adding more extrema is more than countered by overheads at that point. All these methods have similar behaviour, with the timings related to their complexity; from the marginal PCS (channel-wise outliers) to the angular PCS (local hull outliers), all variants must rebuild the measures around merged regions, showing a parallel with the disparity in VAMS and CCS costs seen in section 4.3.

5.3.2 Noise performance

The noise removal ability was tested for all methods. In addition, the effect of noise on segmentation was tested, with the same parameters as before: Gaussian $\sigma^2 = 10^3$ or impulsive $\rho\% = 10$. See next chapter for more detail.

The performance of all methods- including all of the PCS variants- was re-examined: PCSM is the marginal version, PCSV1 is the vectorial one with principal components (PCA), PCSV2 takes the vector from the mean to the region value and then PCA, PCSA is the angle-based version. Like the metrics evaluation, the optimum area threshold was chosen for all, with an area 10 for the uncorrupted case, and moderate Gaussian, impulsive and mixed noise introduced in the image. In this case, the extrema detection in the VAMS method is done using the L_{ae2} metric (see section 6.2.1), due to the slight advantage. Results are shown only for the L_{ae2} rule, with results scaled by a factor of 10^{-2} and NMSE and MCRE shown. See figures D.10 and D.11 in the appendix for a comparison of all sieves at area 100 and 10000.

Filter	Input	Uncorrelated			50% Correlated		
		Gaussian	Impulsive	Mixed	Gaussian	Impulsive	Mixed
none	0	4.621092	11.03877	15.19249	4.602002	7.336491	11.63426
VAMS	0.701100	2.629105	1.136509	3.225414	2.570421	0.746204	2.803816
CCS	0.970979	2.613770	1.349318	3.278503	2.671730	0.955942	2.927100
GS-AOC	**0.560550**	3.206486	1.535769	4.324703	2.478381	0.956966	3.054846
VAMOCS	0.782747	2.643463	1.172610	3.265078	2.606076	0.790099	2.854069
VDMOCS	0.737260	2.864598	1.152242	3.587192	2.782706	0.744941	3.062368
PCSM-o	0.630419	2.250081	1.201541	2.815814	2.202713	0.673666	2.412120
PCSM-oc	0.859053	2.522014	1.304469	3.160123	2.542168	0.852566	2.802272
PCSV1-o	0.588084	2.271481	1.286930	2.797923	2.174306	0.803978	2.405724
PCSV1-oc	0.822746	2.791749	1.486735	3.562829	2.754056	0.998267	3.052828
PCSV2-o	0.684468	**2.200670**	1.094403	**2.712084**	2.166978	0.707527	2.381358
PCSV2-oc	0.867748	2.476292	1.230089	3.126739	2.491765	0.850044	2.740967
PCSA-o	0.637933	2.205576	**1.091301**	2.751061	**2.132431**	**0.653760**	**2.339955**
PCSA-oc	0.788479	2.727839	1.204321	3.473680	2.669101	0.786509	2.965716

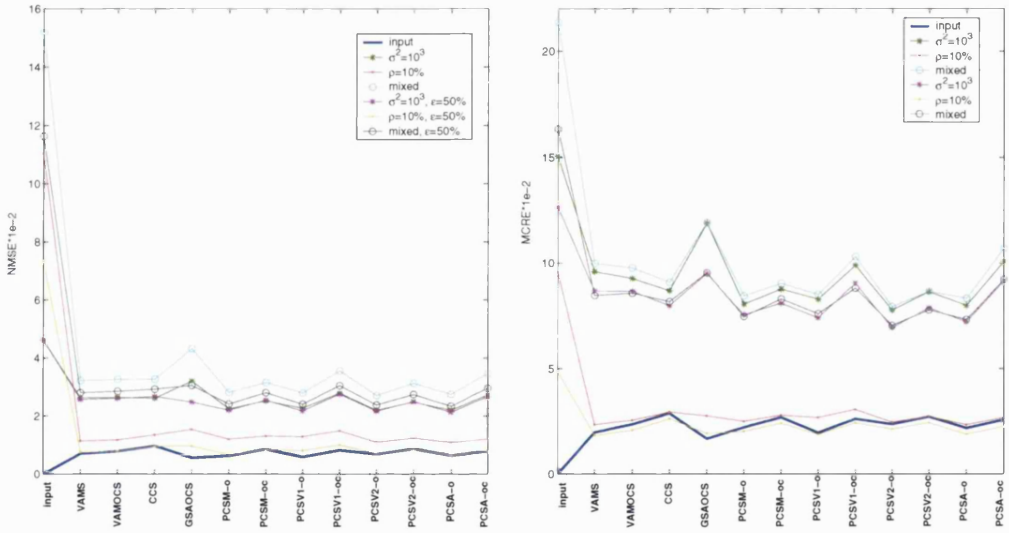
Table 5.1a: NMSE vs MM method

All filters consider correlated and uncorrelated noise in a similar manner, barring the lower initial distortion. The best overall performances are for the newer PCS openings, with the PCA centroid-based (PCS4) and angle-based (PCS5) version being marginally best against Gaussian and impulsive, respectively. That comes at a processing cost, of a factor of 3-4 for the PCS4 and another order of magnitude worse for PCS5. Another fact not shown in the graph is the optimum scale: the best area for Gaussian removal is near 20, higher when mixed with impulsive and lower for all PCS closings. On the same issue, all methods achieve optimum results at similar scales except for the greyscale colour sieve (with a best threshold of 2-5 for any noise models).

The results obtained show once again the better results of the VAMS among the original sieves when the noise present is not impulsive. Another visible point is that the angle-based PCS and VAMS openings introduce the least distortion on the image, and the hull and vector PCS give the most; all as expected, since the first two methods ignore the local edges and plain areas respectively whereas the latter will

Filter	Input	Uncorrelated			50% Correlated		
		Gaussian	Impulsive	Mixed	Gaussian	Impulsive	Mixed
none	0	15.05703	9.564917	21.35214	12.63520	4.839810	16.33831
VAMS	1.970675	9.600471	2.335640	9.990194	8.668185	**1.821043**	8.463807
CCS	2.874030	8.690942	2.938461	9.102737	7.976276	2.614632	8.168680
GS-AOC	**1.664384**	11.88672	2.750847	11.92637	9.495676	1.916940	9.536778
VAMOCS	2.350880	9.278253	2.540622	9.773553	8.658875	2.065674	8.571138
VDMOCS	2.260442	10.38338	2.577905	10.70273	9.327816	1.988708	9.299882
PCSM-o	2.204697	8.051607	2.489599	8.446509	7.545645	2.006600	7.465920
PCSM-oc	2.677818	8.770041	2.774122	9.036365	8.089713	2.391820	8.294127
PCSV1-o	1.952592	8.292920	2.669962	8.517350	7.398625	1.867830	7.598305
PCSV1-oc	2.609609	9.899673	3.055303	10.31113	9.047358	2.417900	8.811049
PCSV2-o	2.356327	**7.769643**	2.451318	**7.937969**	6.921607	2.117520	**7.040141**
PCSV2-oc	2.712290	8.642038	2.731056	8.659167	7.873142	2.425263	7.779050
PCSA-o	2.174686	7.992868	**2.333050**	8.335986	**7.208535**	1.875482	7.324163
PCSA-oc	2.574550	10.09133	2.671262	10.71498	9.151078	2.244484	9.233282

Table 5.1b: MCRE vs MM method



(a) NMSE

(b) MCRE

Figure 5.11: Noise performance for uncorrelated noise

remove both, giving both higher distortion and faster segmentations.

Noise also affects region numbers and timings. In ideal conditions, M-sieves (open-close) are better than openings. But VAMS/PCS openings can outperform M-sieves when Gaussian noise is added, as minima are less representative than in ideal conditions and should be kept whenever possible: the reason being noise induces higher initial extrema counts, especially maxima, boosting speeds yet giving worse pixels to merge to (i.e. minima) overall in both cases.

The Hull method is not affected by the presence of noise, as the number of extrema is already high. Results are less aggressive (and smoother) for VAMS openings, as all other variants and methods do

attack more regions. Impulse noise has little effect on performance, with 1-10% noise per channel only affecting low scale area sieves. The PCS variants are quite different as shown above, making the vector variant a safer choice. The VAMS M-sieves also show similar performances, with the VAMOCS performing better only when Gaussian noise is present; probably due to its link to region contour, which becomes less stable with Gaussian noise.

5.3.3 Conclusions

Compared to both median filters and graph morphology (see chapters 2,3 for noise results), this set of filters have an acceptable performance: similar to vector median filters whilst obeying strong causality and more robust than most greyscale graph morphology extensions to colour against Gaussian noise. This shows that the lower distortion of greyscale sieves is overcome by the lack of usual artifacts (edge jitter and colour creation), the better response against uncorrelated impulsive noise, where mostly only 1 channel is affected at any point, and the increased robustness to Gaussian (slightly lagging the VMF).

Among all methods, the PCS-A achieves the closest results to greyscale, thus achieving its chief purpose, a colour analogue of M-sieves. Its noise performance is among the best, on par the PCS-V. However, its large running times make it rather unpractical, leaving the PCS-V and VAMS as the main options for low scale processing.

Chapter 6

Colour morphology studies

The sieving options so far examined have been based on the use of different outlier definitions and processing (changing the merging order), giving all the sieves seen before. Other options that graph morphology offers are attributes (merging order), distance metrics (merging criterion) and the face value of the merged region (merging model).

6.1 Attributes

One of the main parameters in attribute morphology is the attribute used. Classic greyscale attributes are area, power, volume and contrast, all of them increasing i.e. region growth always increases its attribute. Less common increasing attributes are the moments of inertia and the dimensions of the enclosing rectangle; for non-increasing attributes, normalised inertial moments [150] and other geometric constraints (compactness, skeleton length, perimeter) are common examples.

Here only generally increasing attributes are studied, so the basic sieving method seen in section 3.2 for openings can be used. Note that non-increasing attributes, although possible by using tree post-processing, would give similar problems as found by [56] i.e. whether to stop before or after passing the threshold, and if to resume merging for any region beyond the threshold at any time. In addition to area, the main attributes used were colour extensions to contrast, hull volume, volume and power.

Also some research was aimed at multiple attribute sieving. This is partly related to vector attributes; Wilkinson et al. take thinnings within some distance from the target attribute [151], whereas here openings proceed to the maximum scale, like marginal attributes.

Despite the large volume of region-growing and attribute-based research, the study of colour attributes is still basic beyond some extensions to contrast [29], coherency [66] or entropy. A set of unambiguous, universally accepted definitions has not been determined, hence different definitions are available.

6.1.1 Contrast

Contrast is defined by the points selected among the neighbouring regions along with all prior values within the region of interest. In essence, here contrast takes the maximum distance among all internal points and the external ones near, if not within, the internal range. As the extension of some of the greyscale attribute definitions involving nearby regions to colour are unfixed, a range of options were considered, with contrast as an example:

- The internal contrast, derived from the points within the region, as in greyscale. That definition implies initial contrast is zero and therefore allows the removal of impulsive noise, making it more aggressive than greyscale. However, since the attribute only depends on its region and is increasing, this was the initial implementation, and is the version used for multi-attribute sieving.

- The external contrast, among the points in the region and its boundary regions, minimizing the external distance; the choice against the most different neighbour can be supported by treating contrast as the largest step possible before absorbing another region, rather than the largest step without needing to absorb any region. This is a more complete definition, based on Heucke et al. [23], who also consider object and background brightness with some success, and vaguely on Salembier et al. [132], who deal with mean variance to the background rather than distance to it, and an analogue to greyscale when sieving up to a threshold rather than beyond it, despite a worse response against impulsive noise.

  However, this is a non-increasing criterion in a strict sense: although the merged regions increase their metric, their neighbours may decrease it as a result of the changes, so region attributes are interdependent. By taking the distance to the neighbours rather than just the internal one, this achieves a minor improvement in noise and segmentation performance over the previous definition in most cases.

- The full data range (see figure 6.1) maximizing the distance between all points in the region to those in the neighbours. Despite being increasing this even less robust to noise, goes beyond the classic contrast definition and cannot be applied to other attributes.

As an example, consider the central region (with face values marked with **X**) in figure 6.1. The internal contrast (1) is the full range of values within the region. The external contrast (2) takes the internal contrast and the closest neighbour vector to it; since the left region has a face value (**X**) within range, the external and internal contrast are identical. The full data range (3) spans all data, not only face values, in both adjacent regions, surpassing other definitions.

Although research has been initially limited to increasing and stable attributes, the first two types of contrast were selected. They both have efficient, logical and quite similar segmentations despite not having a perfect definition; besides, contrast is only idempotent when a reference to its limits is available with the sieved image, so the differences expected are not important. Three further subdivisions were made in contrast as seen in figure 6.2:

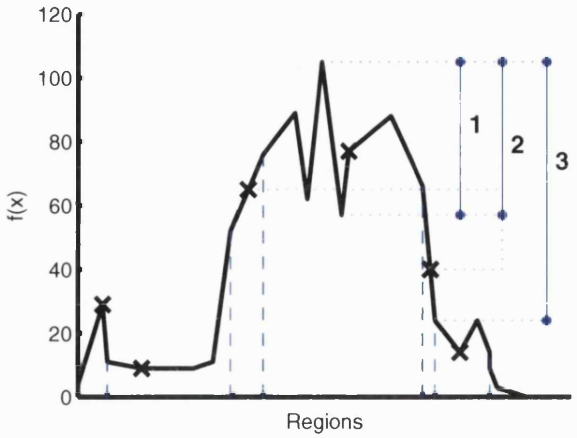


Figure 6.1: Contrast types: internal (1), external (2) and full data range (3). Face values marked.

- line contrast: defined as the internal contrast, the largest distance between any two points within the region hull. Three distance rules were considered: city distance, Euclidean and angular-Euclidean (this gives a higher importance to chroma changes). Extends up to 765 (L_1), 442 (L_2) or 100 (L_{ae1}) for the RGB colour space model within the range (0,255).

  After including the surrounding region values there was a slight improvement in segmentation and smoothing (i.e. less distortion), yet with worse results noise-wise. In this case, the distance is the minimum of the maximum distances from the neighbours to the hull; if any is within the local hull, the distance is less than the internal contrast, so the latter is taken.

- hue contrast: the distance between the hue extrema in the range containing the region hue, or the overall hue range to each side of the region hue. Extends to 180° or a half circumference of maximum difference between any 2 points (fig. 6.2b), or to the full range of 360° (fig. 6.2c) as 180° on either side of the region vector [7].

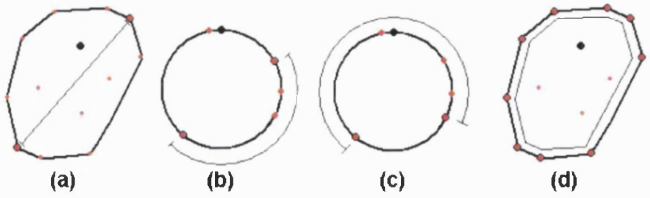


Figure 6.2: Contrast definitions:(a) line (b) half-range hue (c) full-range hue (d) volume contrast.
Region face value is marked in black, neighbours in red.

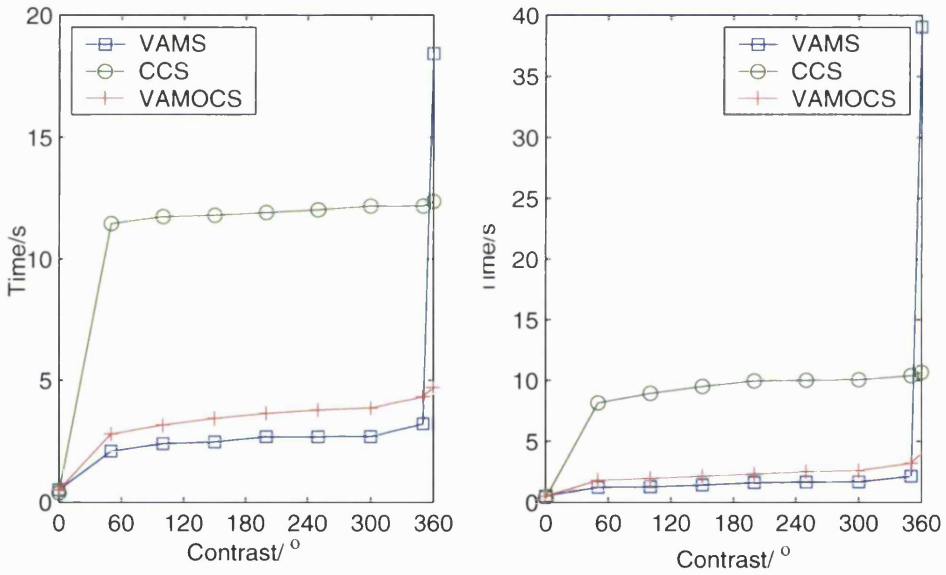
- hull volume: the volume enclosed by a 3-D hull, or the area by a 2-D hull. Technically this is done by the hull algorithm used[146], so it is limited to colour. This proves superior to line contrast segmentation-wise (see chapter 7) as variations within more than one direction are detected. However, defining the minimum volume among multiple regions, especially that for overlapping

hulls, proved too complex for practical implementations; in all other cases, the variant using just all face value references to other regions again gives better results, at much higher calculation costs.

- hull perimeter: the perimeter, or area in 3-D, of the hull. A downgrade from hull volume, but otherwise similar in performance and calculation costs.

Initial noise results for the line contrast also showed clearly lower speed at high scales, openings outperforming other sieving forms with timings an order of magnitude higher than area sieving. This was due to the method checking (M*N) points at merging, where M and N stand for the size of the regions, favouring the merging of unequal regions in openings. This was solved by constructing a hull of the points for any region; this reduced timings at larger sales, giving a factor of 2-3 in speed for all contrast sieves to high scales.

As in the case of power, performance is linked to the number of steps: contrast sieves using the L_{ae} metric double the speed of the L_2 metric contrast in figure 6.4 and the hue (360°) contrast in figure 6.3 again proves faster except for the VAMS, probably due to large minima. The hull volume contrast proved superior in other aspects to the other versions in most cases.

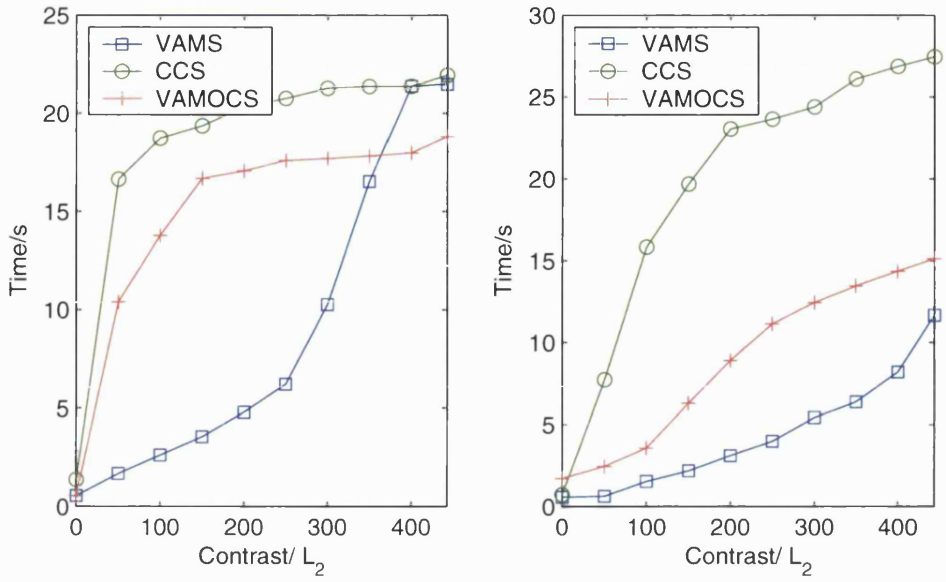


(a) no noise (b) mixed noise

Figure 6.3: Timings for hue contrast sieving, 'lily' image

6.1.2 Power

The power is defined as in greyscale [138], as the collective power of all pixels in the region to the region vector, but with one difference. Since greyscale power only considers 1 component per pixel, the extension is not obvious but trivial: the squared magnitude of the vectors between the region's value and



(a) no noise (b) mixed noise

Figure 6.4: Timings for line contrast sieving, 'lily' image

its composing pixels. This is equivalent to taking the square Euclidean metric of the vectors, or simply their total marginal powers. This is unbounded and generally increasing: the final power depends on the image contents and the face values chosen, and power for non-outliers always increases. Given a region size N with face value $\vec{Y} = \{Y_1, Y_2, \ldots Y_D\}$ and point values $\vec{x}$,

$$\sum_{n\in N} ||\vec{x_n} - \vec{Y}||_2 = \sum_{n\in N} \sum_{d=1}^{D} (x_{n,d} - Y_d)^2 = \sum_{d=1}^{D} \sum_{n\in N} (x_{n,d} - Y_d)^2 \tag{6.1}$$

$$\Rightarrow \text{vector power} = \text{sum of marginal powers}$$

Performance times are again slightly higher than in area sieving, yet as in figures D.5-D.7 the visual performance is much better, being best suited for smoothing. It also means that most outliers are kept, thus being weaker against impulsive noise: a 10^5 power sieve is needed to remove impulsive noise, or 10^4 to remove noise to a moderate level, compared to a scale 10 area constraint. The performance on Gaussian noise mimics that of the area openings, both being generally weak against it.

The main feature introduced was measurement of power from another reference vector. This was introduced to counter the effect of outlier and metric definitions other than luminance sieving, i.e. the change or decrease in overall power, given a new face value. Note this makes power non-increasing just like external contrast, and can occasionally happen in greyscale sieves with nested extrema. Three options were studied: volume to the marginal median, to the mean and to the region's face value, i.e. the greyscale equivalent. From the latter, only power to the mean proves strictly increasing. Results prove that attributes to a centroid are more predictable in outcome and more uniformly increasing, yet cannot reach the same peak performance or attribute values.

6.1.3 Volume

Volume was defined as the city-distance metric volume of the set, in line with the definition of power; the same options for a centroid were also studied. Properties are also similar to power: it is strictly increasing for the median centroid, with a moderate increase in processing times respect to area.

6.1.4 Other attributes

Other attributes examined were:

- Entropy: An experimental one, based on its greyscale definition [125]. This involved a rough clustering using code from [152], hence giving the entropy of a pixel over the image. The results proved similar to area, so it was discarded.
- Surface: A combination of area and gradient, or the actual surface of the region for greyscale. Results were not conclusive, so it was abandoned too.
- Inertia: The moment of inertia of the pixels around the center of the region, or of the pixels and value vector, as in power. Also similar to area, but with a closer link to power.

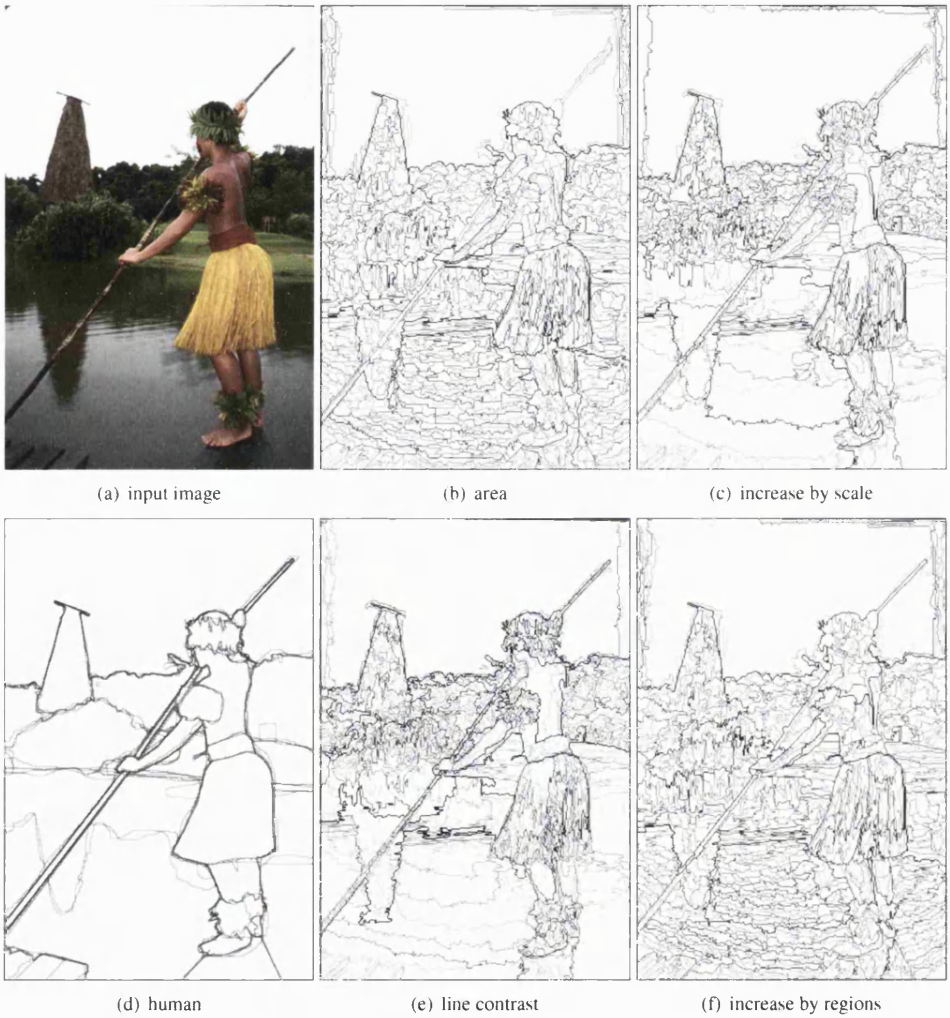
6.1.5 Multi-attribute sieving

Finally, combined attribute sieves were developed; this involved sieving to different attribute limits, overcoming the limits of each other, e.g. sieving to area and contrast for a more uniform denoising/ simplification. Although the overall processing order does not significantly alter the end result with fixed thresholds, it does affect the middle stages in a multi-step approach and all the results when sieving to a region number. Since new extrema are generated, unlike in greyscale, the results are affected by the order of the sieves; figure 6.5 gives an example with contrast (highly value dependent) and area (independent of value, texture), segmented at equal levels between 1000 and 1 regions. Three options were considered:

- Sequentially, e.g. contrast then area opening. This does not allow a single limit, as in region count.
- Alternating by scale, or proportional attributes. The attribute values are normalised by the maximum value feasible or the image (or the scale chosen), and attributes are alternated so as to keep all at the same proportion of their maximum. For example, sieving an image of size 10000 and maximum range 256, sieved to area 1000 and contrast 50, gives 1 area step as .1% and 1 contrast step as 2%. Thus the area threshold increases by 50 for every contrast step of 1, assuming integer step values. As seen in figure 6.5c, the attribute with fewer steps, line contrast (figure 6.5e), produces the main changes.
- Alternating by regions, or least increase. As in the last option, the attribute values are normalised, but here attributes are chosen so as to force the least change, by sieving the least extrema possible. Since this cannot be known beforehand, the number of regions is taken instead. For example, if the number of regions with a given area is .01% and the number with a given contrast is .5%, then area

sieves take precedence. As in figure 6.5f, there is more balance between both attributes, with the effects of contrast being more evident at low, darker segment numbers (e.g. the staff, the texture within the skirt) while at finer segmentations area (figure 6.5c) dominates.

In all cases sieves are looped to maintain idempotency for all attributes at each iteration. The use of multiple attributes noticeably increases timing. Like in classic sieves, region numbers can be used as a sieving threshold, letting it choose the best threshold combinations. For the region alternating scheme this gives the closest method to the vector attribute, since attribute increases are fairly independent of each other, considering the correlation between increasing attributes [151].



(a) input image (b) area (c) increase by scale

(d) human (e) line contrast (f) increase by regions

Figure 6.5: Multiple attribute sieve example for area and contrast: (a) input image (d) human segmentation by 5 subjects (b)(e) CCS sieves (c)(f) CCS multi-attribute sieves

6.1.6 Noise Evaluation

Noise evaluation results follow in tables 6.1- 6.4. The CCS was used for the 7 test images and different attributes; the scale chosen for the uncorrupted input was the average scale for best noise removal for that attribute. The scales chosen for area and contrast were unit steps to a scale of 50, with exponential steps from that point; for all other methods exponential steps of $10^{1/2}$ were used since their limits, opposite to their sensitivity, are $10^2 - 10^5$ greater than for area.

Attribute	Input	Uncorrelated			50% Correlated		
		Gaussian	Impulsive	Mixed	Gaussian	Impulsive	Mixed
none	0	4.602002	7.336491	11.63426	4.621092	11.03877	15.19249
Area	1.134799	2.613770	1.349318	3.278503	2.671732	0.955942	2.927100
Contrast	18.02670	2.449531	2.645097	3.521118	2.540576	2.671381	3.187936
Power	1.287711	2.580223	1.647388	3.314351	2.601704	1.211446	3.045722
Volume	1.368168	2.580108	1.459066	3.278323	2.622523	1.123555	2.959238
V.Cont	5.071228	2.547440	1.343107	3.297891	2.580850	0.979103	2.993531

Table 6.1: NMSE/10^-2 for attributes in CCS

Attribute	Input	Uncorrelated			50% Correlated		
		Gaussian	Impulsive	Mixed	Gaussian	Impulsive	Mixed
none	0	15.05703	9.564917	21.35214	12.63520	4.839810	16.33831
Area	3.068248	8.690942	2.938461	9.102737	7.976276	2.614632	8.168680
Contrast	9.849773	7.944723	4.313863	8.204000	7.085655	4.428207	6.571638
Power	3.405343	8.871680	3.325969	8.596843	7.827264	2.304647	7.156054
Volume	3.484585	8.909965	3.181679	8.921191	7.986468	2.354985	7.435375
V.Cont	6.144565	6.875775	3.025646	8.622887	7.119821	2.774205	7.531917

Table 6.2: MCRE/10^-2 for attributes in CCS

In terms of results, area is the best attribute; note the large distortion caused by contrast on the original image. This is mostly due to impulsive noise, which halts sieving until scales where all features are removed in the original image; a CCS contrast sieve to the scale where only Gaussian noise is removed gives an NMSE of 4.01, compared to 18.03 for impulsive. Volume behaves like the area constraint: it is increasing although non-conservative, with superior results only for mixed noise. Power proves even less similar to area; this, together with its dependence on pixels values, makes it inferior against long-tailed noise, yet gives the least distortion in ideal conditions as seen in figures 6.6 and 6.7 for all attributes, to a similar degree of simplification in each method; although different methods and scales are used, it is clear power has the least effect on the image, and taking volume from a centroid (fig.6.7d) has a greater effect on the amount of sieving than with power (fig.6.7b) while being even more similar to area.

Contrast is different from the above: a region can grow without increasing its metric by taking points within its rank, removing all detail within range, especially Gaussian distributed. In greyscale this is prevented by neighbouring extrema; in colour it can do so by increasing the contrast in other directions, or by a decrease depending on neighbouring mergings, what explains the sudden growth at higher scales in figures D.2-D.4 compared to power in D.5-D.7 or area. Hue contrast (not shown) proves only adequate

Attribute	Input	Uncorrelated			50% Correlated		
		Gaussian	Impulsive	Mixed	Gaussian	Impulsive	Mixed
Area	0.799865	2.864598	1.152242	3.587192	2.782706	0.744941	3.071538
Contrast	18.12414	2.794700	2.516202	4.132966	2.767354	2.280029	3.378184
Power	0.176123	2.953269	2.221965	3.728588	2.884802	1.133974	3.131990
Volume	1.084396	2.894488	1.252369	3.609429	2.755131	0.966846	3.070686
V.Cont	17.27702	2.851630	1.804799	3.612597	2.724607	0.958513	3.107916

Table 6.3: NMSE/$10^{-}2$ for attributes in VDMOCS

Attribute	Input	Uncorrelated			50% Correlated		
		Gaussian	Impulsive	Mixed	Gaussian	Impulsive	Mixed
Area	2.380859	10.38338	2.577905	11.19369	9.327816	1.988708	9.547036
Contrast	10.20582	10.04666	4.316757	11.05872	9.069350	4.141722	8.549080
Power	0.532764	10.43165	3.754541	9.494192	9.275887	1.343752	8.662607
Volume	3.044172	10.46905	2.754973	10.84003	9.209265	2.302254	8.916534
V.Cont	12.60373	10.02878	3.490648	10.76763	8.778283	1.564873	9.265150

Table 6.4: MCRE/$10^{-}2$ for attributes in VDMOCS

provided the thresholds are well chosen, since its undefined behaviour for low and high luminance values prevents proper merging for large attributes (see results in appendix D.4).

These features of contrast lead to problems in sieving and segmentation not discussed before, namely whether to stop merging before or after exceeding the threshold and whether or not to grow those above threshold until this happens elsewhere. A method to counter these is the use of sieve trees for segmentation, which is explored in the next chapter.

(a) area sieve to 150 (b) power sieve to $5*10^5$ (c) L_2 contrast sieve to 150 (d) volume sieve to $1.5*10^4$

Figure 6.6: Attribute GSAOCS sieve comparison: power, area and line contrast

(a) area sieve to 300 (b) power sieve to $5*10^6$ (c) L_2 contrast sieve to 300 (d) volume sieve to $3*10^4$

Figure 6.7: Attribute VAMS sieve comparison: power, area and line contrast

In conclusion, area opening is the best attribute for noise removal due to being independent from pixel

values, whereas contrast sieving proves best at low noise amplitude levels and gives the best simplification for uniform images, see figure D.2. Power is the most gradual in changes, smoothing out data with few visible effects up to high values. Mixed attributes are another option if several properties are of importance.

6.2 Metrics

The methods above require distance metrics, for VAMS extrema detection and for selecting the best neighbour when merging to extreme in all methods. The best neighbour selection is done by taking the lowest metric to the extrema, followed by their luminance difference and G,R,B differences if equal pairs are found as in [142], not raster scan as in [143]. The rules used affect the results; it should be noted that in the VAMS methods the same metric need not be applied to the extrema detection and neighbour selection stages.

6.2.1 Distance metrics

Two types were used:

- magnitude: L_1, L_2, L_2^2, Y. These take the vector magnitude between two points. This is known to perform well in uniform spaces, like Lab, where equal distances mean equal perceived differences over short distances. For RGB, the sensitivity to hue is low.
- direction: angular (or spectral angle measure) or L_θ, angular-Euclidean or L_{ael}. Take the vector angle between 2 points. Since both hue and saturation changes can be described with an angle, this is best for colour spaces. The angular metric is the angle between 2 vectors except for null vectors, as seen in Chapter 2. That is,

$$L_\theta(\bar{x},\bar{y}) = \begin{cases} \sum_{j=1}^{D} \arccos(\frac{x_j \cdot y_j}{|x_j||y_j|}) & \forall |x|, |y| \neq 0 \\ 0 & \forall |x| \equiv |y| \equiv 0 \\ \sum_{j=1}^{D} \arccos(\frac{x_j \cdot 1 + y_j \cdot 1}{\sqrt{D}(|x_j|+|y_j|)}) & \text{otherwise} \end{cases} \tag{6.2}$$

  considering **1** as the unit vector. This gives a measure of saturation and hue difference, but without regarding the magnitude of the vectors.

The angular-Euclidean rule is first described [153], which resembles that in [154]. This takes the angular and Euclidean, and combines as:

$$L_{ael}(\bar{x},\bar{y}) = 1 - (1 - L_\theta/(\pi/2))(1 - L_2/(255 * \sqrt{3})) \tag{6.3}$$
$$L_\theta(\bar{x},\bar{y}) = \bar{x} \cdot \bar{y}/|\bar{x}||\bar{y}|$$

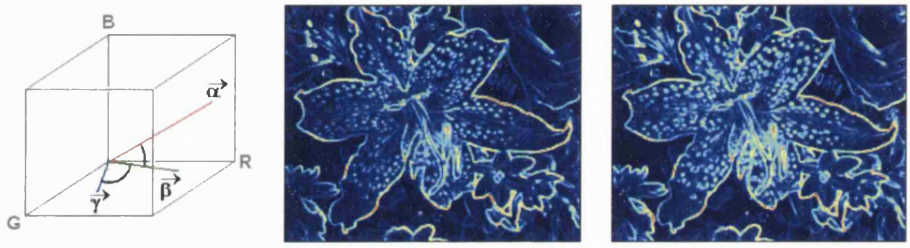
This gives an Euclidean base with a boost in hue performance, especially for low luminance and small distance differences. Another variant was developed, which removes the problem of hue influence at low

intensity values by taking the angle from (R,G,B)=255 if this smaller, see [153] or [102]:

$$\begin{aligned} L_{ae2} &= 1 - (1 - L_{\theta m}/(\pi/2))(1 - L_2/(255 * \sqrt{3})) \\ L_{\theta m}(\bar{x}, \bar{y}) &= min(L_\theta(\bar{x}, \bar{y}), L_\theta(\bar{I} - \bar{x}, \bar{I} - \bar{y})), \bar{I} = [1, 1, 1] \end{aligned} \tag{6.4}$$

This metric gives the smallest angle from the black or white points, and hence gives more importance to saturation than to hue when either vector is near the extremes, giving preference to hue when they are mid-valued; the only drawback is that 4 rather than 2 points with equal hue can take the same distance.

A graph for the L_{ae} space and the gradient for L_2 and L_{ae} rules follow in figure 6.8. It can be seen that vectors $\vec{\beta}$ and $\vec{\alpha}$ are nearer than $\vec{\beta}$ and $\vec{\gamma}$ in terms of angle, yet have similar Euclidean distances, so $\vec{\alpha}$ should be preferred over $\vec{\gamma}$. This means the L_{ae2} measure helps to pick out extrema, and combined with its sensitivity to hue made it the preferred rule for most results. As a proof, the number of initial VAMS maxima in the image shown was 2155 for the luminance metric, 2118 for the Euclidean and 2209 for the L_{ae2}, a minor advantage for extrema detection and merging with uses for noise removal. Also, figure D.10 shows the effects of the L_{ae2} rule on all colour sieves; in all cases, especially in the presence of noise, the hues are closer to the initial ones than in figure D.11, although with a small penalty to edge definition at higher scales.



(a) L_{ae1} metric (b) VAMS field,L_2 metric (c) VAMS field,L_{ae2} metric

Figure 6.8: L_{aex} configuration and examples

On a related topic are the results of colour space transforms. Using a different colour space has no visible effects on sieving an uncorrupted image; including widespread noise gives a severe degradation in terms of performance. An example is in Lab for uncorrelated noise, where chromatic noise persists while luminance, and hence non-corrupted hues, is more coherent; that improves the NMSE but affects the MCRE. In HSI spaces it is worse, since one of the components is cyclical; splitting hue in separate components (like in Lab) improves the results, but still leaves the problem of over/under saturated tones.

All this seems to suggest it is best to use the space where noise is less correlated, which often is the initial one. The effect of different spaces is less for the VAMS sieve than for the CCS/PCS, where the key factor is mainly position; that changes for spaces with quite different ranges for each axis, clearly affecting the VAMS field. An example is VAMOCS and CCS segmentations (see chapter 7): VAMOCS and CCS performances are closely on par, with small decreases in segmentation goodness for the Lab space. If Gaussian noise is added, the VAMOCS score quickly drops in the RGB space, while switching to Lab gives equally worse scores for both methods.

6.2.2 Geometric/energy metrics

Classic metrics involve only the distance between regions. This ignores the relative shapes, and those of other regions. New metrics were developed for such purpose, inspired by other authors [155] [75]. The aim is taking shape information into account.

The metrics considered were merging to another reference point (median/mean) or with added constraints. The latter involve minimising a classic distance and other parameter (be it the perimeter-area ratio (P/A), P/A change, or total distance change). Another option were minimising the VAMS of the resulting regions (being energy related) and an adaptation of the Earth Mover Distance (EMD) [156]. These were:

- normalised range metrics, or combinations such as product of channel differences or offset differences.
- geometric constraints
- Earth mover's distance (EMD)
- Energy snakes

6.2.2.1 Normalised range metrics

One of the disadvantages of the metrics seen is they are often geared to one type of noise, which is the main weakness of the L_2 norm. A metric that rewards close matches in most channels while picking the nearest point in space hence was deemed useful in terms of impulse removal.

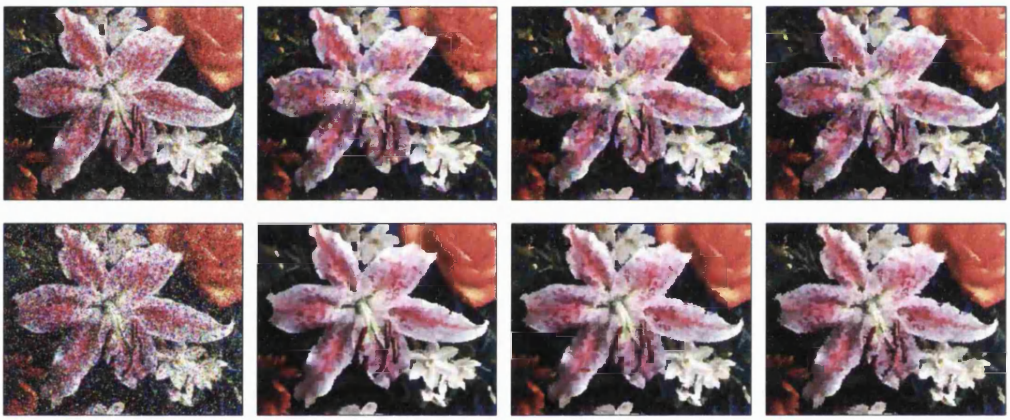
The first option was to decrease the output to favour this. Taking G as a constant and x as the channel-wise difference, then using $f(x) = 1 - \frac{1/G}{(x+1/G)} = \frac{Gx}{(Gx+1)}$ for all channels gives a finite range independent from x and bound between 0 and 1, and had the right behaviour, i.e. close matches in multiple channels are favoured and larger differences in a channel are allowed than in the L_1 or L_2 norm. G determines the mismatch allowed, but constant values within 1-10 divided by the range (255 in this case), or dependent on the actual differences, give acceptable results.

In the same trend goes the product of channel differences, or offset differences; this shares the properties of the L_1 metric against impulse noise, rewarding close matches in one or more channels. Different options were considered: $\prod^N(\delta_c)$, $\prod^N(1/\delta_c)$, $f(\prod^N(\delta_c)+\delta_c^N)$, and $\sqrt[x]{\sum^N(\delta_c)^x}$, where δ_c is the channel-wise difference between inputs and may be offset to avoid division overflows. Their impulse removal is effective, but a common side effect is merging of fairly different regions, especially at higher scales. Such metrics are also not ambivalent in the VAMS: finding the best fit to remove impulse corruption conflicts with its detection by that metric distance and relies on other metrics for proper extrema identification, what explains the problems associated with the GSAOCS in most cases.

The overall results were mixed, with a bias to given noise types. For the normalised range metric, using increasing and fixed values of G quickly degrades the performance against Gaussian while adaptive values, such as $G = 255 * \max(\delta_c)$, decreases the dependency on image contents; even then, noise results are close to those by luminance metrics (see section 6.2.3). Among the product metrics, the merge metric $min(\delta_i\delta_j + \delta_k^2)$, given $i \neq j \neq k$, improves results over L_1 for impulsive noise by about 3% in all metrics

on the standard image set when used in the VAMS, but has otherwise average or below average results; the opposite metric, $max(\delta_i\delta_j + \delta_k^2)$, responds well to Gaussian and works equally well in any sieve, but creates severe edge distortion when removing impulsive noise. Likewise, the metric $\prod_i^N(\delta_i)$ fares much better against impulsive noise.

One of the better metrics examined is the product of differences, $\sum_{i=1}^{N}\sum_{j=i+1}^{N}(\delta_i\delta_j)$. Results have little or no difference compared to the L_1 or L_2 metric on Gaussian noise, but have a key effect on pure impulsive noise- 5 to 10% decrease in filtered NMSE respectively for the CCS and VAMS and smoother boundaries. Also the $L_{3/2}$ metric, or $(\sum x^{3/2})^{2/3}$, proves somewhat useful, ranging between the L_1 and L_2 in results as seen below, for 20% impulsive and Gaussian variance of 2000; this metric (fig. 6.9c) keeps smoother boundaries than the Euclidean for impulse corruption (its main weakness) while still selecting similar hues for either type of noise, which is an advantage over the L_1 metric except for impulsive noise. However, and despite the latter results, considering only the distance between region values does not seem the best strategy to improving the colour sieves.



(a) input (b) L_1 (c) $L_{3/2}$ (d) L_2

Figure 6.9: Comparison between sieve metrics on Gaussian (top) and impulsive (bottom) noise

6.2.2.2 Geometric-Energy constraints

Three sets of constraints were considered: geometric constraints, the EMD and active contours.

Geometric constraints use geometric ratios and distance metrics and minimise both, or their combination. This was a quick alternative to energy metrics, and were latter combined. Generally most constraints prove less than useful. The main exception is the change in the perimeter-area ratio (P/A) or complexity ratio (decreasing with size, and ranging from 0 to the connectivity size) especially if combined with the metric distance (better if normalised to total input P/A), even in this case the distance metric proves best. This works by giving a bonus to removing smaller regions, giving more regular segmentations.

Most others, as perimeter/common perimeter ratios or contact ratio (increasing with amount of detail, within 0...1 range) or convexity as perimeter per neighbours (increasing with number of corners and size, and ranging from 1 to the connectivity size), achieve less adequate results.

The Earth Mover's Distance (EMD) [156] is a graph/set theory concept, corresponding the minimal cost of a set of flows joining sources and sinks, or filling 'holes' with 'earth' with the least effort/distance. For regions, the cost for regions with a large difference, in face value or position, or reduced contact area should be greater, or the cost 'flooding' a region should consider more than gradient.

The metric used was a metric distance with a shape measure. This measure is given by the geometric mean of areas and distance to new region centre, normalised by the proportion of shared perimeter; minimizing the metric then should favour the removal of smaller or complex regions.

Results showed a slight improvement; using the ratio of common to uncommon perimeter length gave a very slight benefit (a metric improvement of 1%), but using the average distance to other neighbours or the actual areas again proved counterproductive.

Finally, energy methods are similar to the EMD, but use the snake minimisation to pick roughly convex regions: natural regions tend to be so [75]. Other of their options is minimising distance to region centroids (mean/median).

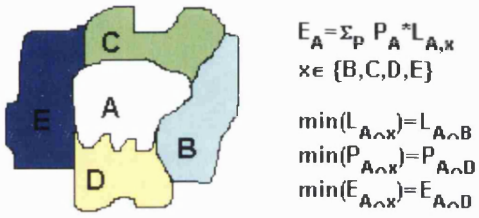


Figure 6.10: Illustration of energy metrics

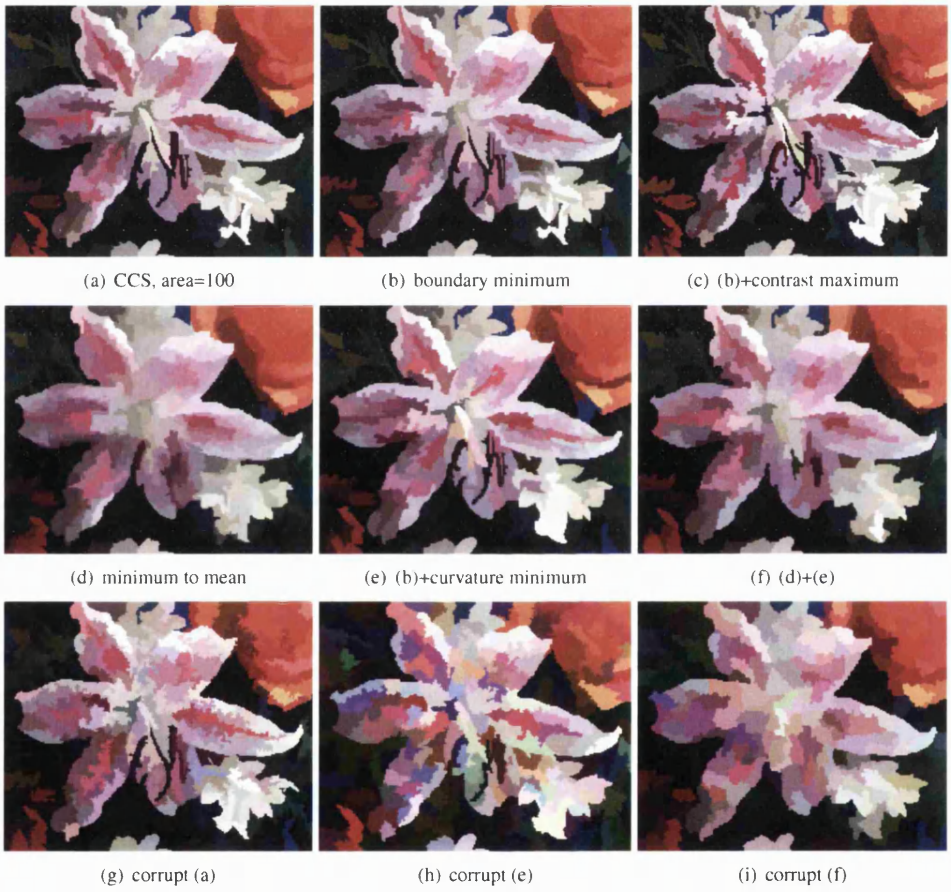
The energy metric used [157] works as follows:

- the energy of snakes is partly internal and external, with different weights. The target is to minimise the total energy after any merge. In figure 6.10, that corresponds to removing region *D*, and likewise if only considering the shape; following distance metrics would favour *B* instead. Rather than taking inner or outer pixels (dependent on region dimensions), the middle path is taken: unit segments among the 4-connected boundaries, which is present even when the region has unit width or reaches the image boundaries (with some problems). 8-connected boundaries would add more complexity and are ignored. External (gradient) energy is taken at the actual mid boundaries, subtracted from the internal energy and assumed zero at the image borders.

- In [157], the curvature is taken from 3 adjacent points and the regularity from 3 curvatures/5 points. Here, curvature is averaged for adjacent opposite angles. For simplicity, boundaries are extracted taking the shortest closed path. This gives multiple snakes per surface (a must when a region is the background) and penalises sparse connections.

- Timings are somewhat higher, especially when taking the actual contour energy, rather than the approximate one (to the average neighbour).

The results in figure 6.11 give moderate segmentation improvements, with the visible effect of more compact regions when minimising perimeter, curvature and, unlike classic snakes, contrast (fig. 6.11b,e,c); ignoring the latter gives sharper, more relevant boundaries but degrades the overall appearance, as seen by comparing the closest to all neighbours or the mean (fig. 6.11d) and the one farthest from its neighbours (fig. 6.11c). In terms of robustness (section 7.2.1), including the mean factor gives better tones, and denoising, at the cost of rounding borders and removing more features (fig. 6.11h,i). Overall, the best options segmentation wise are merging to the mean of all neighbours, minimising the P/A ratio and vector difference in other metrics. The EMD may be superior, that is depending on the implementation as seen in using different energy metrics. That probably gives a incentive to multiple measures yet to avoid geometric constraints if possible (at least those examined).



(a) CCS, area=100 (b) boundary minimum (c) (b)+contrast maximum

(d) minimum to mean (e) (b)+curvature minimum (f) (d)+(e)

(g) corrupt (a) (h) corrupt (e) (i) corrupt (f)

Figure 6.11: Example of geometric metrics for CCS

Calculating the full energy (curvature and length) has more visible results, with the usual tradeoff between region values and edge stability. The actual weighting used, $F = L_{x,y} * (1.1 - E_{x \bigcap y})$, allows for some tuning. Setting the 1.1 factor to 1 or 1.5 visibly changes the influence of shape, from forcing completely regular regions to having no effect, respectively, although it is restricted to the CCS. In edge-

based techniques (VAMOCS, VAMS), the choice of sharp extrema appears to counter the weighting, giving no clear benefits.

6.2.3 Noise Evaluation

The algorithms studied depend on a best-neighbour replacement strategy to grow regions. Since the choice of a distance metric does have an effect on which pixels are merged, this variable was investigated on its own in terms of noise reduction performance. The metrics chosen were the city-block, Euclidean, angular, the two variants of the angular-Euclidean rule seen in this section (L_{ae1} and L_{ae2}), and the Euclidean/ city-block hybrid or $L_{3/2}$.

The VAMS, the CCS and the luminance GS-AOC sieves were used for evaluation, since they are the least similar in merging behaviour. The VAMS depends on the use of a metric distance for extrema detection, so the VAMS gradient metric was L_2. Results shown are the average for the standard image set from appendix D.1, with identical noise settings, sieved to area 12 or to the minimal NMSE level, with NMSE and MCRE shown for each setting of uncorrelated Gaussian $\sigma^2 = 1000$, impulsive $\rho = 10\%$ and mixed noise. All results are scaled by a 10^{-2} factor.

Table 6.5a: NMSE/10^{-2} to scale 12 or optimum for the CCS

Metric	Input	Gaussian	Impulsive	Mixed
Input	0	4.624677	11.038768	15.205812
L_1	1.051891	2.758314	0.995236	3.302222
L_2	1.043979	2.614192	1.351184	3.292736
L_γ	1.202964	4.624677	2.412253	7.052220
L_θ	2.698874	4.624677	2.596640	8.626439
L_{ae1}	1.107089	2.898142	1.378438	3.597724
L_{ae2}	1.159031	3.367018	1.432304	4.121935
$L_{3/2}$	1.041297	2.642273	1.183587	3.238391

Table 6.5b: MCRE/10^{-2} to scale 12 or optimum for the CCS

Metric	Input	Gaussian	Impulsive	Mixed
Input	0	15.059435	9.564917	21.377353
L_1	3.038083	9.728892	2.807976	10.115815
L_2	2.965276	8.592924	2.937962	9.027052
L_γ	3.612470	15.059435	4.469423	16.606416
L_θ	2.530309	15.059440	2.750044	10.332808
L_{ae1}	2.867133	8.556830	2.836472	9.760507
L_{ae2}	2.675223	8.248748	2.753347	8.595040
$L_{3/2}$	2.985465	8.808811	2.884426	9.307384

As seen in figures 6.12-6.13, the choice of the best metric depends on the main type of noise. It can be seen that the angular rule is the worst of all, always providing a higher NMSE than that of the image itself, yet its MCRE has an acceptable value; a far second is in the luminance rule, with the worst MCRE amongst all. The first effect is due to the angular metric insensitivity to luminance changes and their relevance in the images chosen, which shows their importance for the NMSE. The second one is due to

Table 6.5c: NMSE/10^{-2} to scale 12 or optimum for the VAMS

Metric	Input	Gaussian	Impulsive	Mixed
Input	0	4.624677	11.038768	15.205812
L_1	0.762175	2.708971	0.805264	3.195934
L_2	0.754737	2.620291	1.136560	3.229885
L_Y	0.858178	3.446614	1.516886	4.361977
L_θ	2.223015	4.624677	2.148824	6.142912
L_{ae1}	0.804094	3.322167	1.177618	3.870234
L_{ae2}	0.874679	3.213243	1.204009	3.850323
$L_{3/2}$	0.749218	2.632331	1.148456	3.257339

Table 6.5d: MCRE/10^{-2} to scale 12 or optimum for the VAMS

Metric	Input	Gaussian	Impulsive	Mixed
Input	0	15.059435	9.564917	21.377353
L_1	2.112532	9.957259	2.170222	10.597665
L_2	2.062651	9.526229	2.335560	10.046079
L_Y	2.504024	12.126723	3.005044	12.422793
L_θ	1.884534	15.059435	2.354757	12.161045
L_{ae1}	1.981410	11.238626	2.383290	11.085764
L_{ae2}	1.918903	9.348361	2.312672	9.608238
$L_{3/2}$	2.053567	9.572871	2.356924	10.160722

Table 6.5e: NMSE/10^{-2} to scale 12 or optimum for GS-AOC

Metric	Input	Gaussian	Impulsive	Mixed
Input	0	4.624677	11.038768	15.205812
L_1	0.631953	3.017875	2.846287	4.251210
L_2	0.632803	2.914794	3.018066	4.232001
L_Y	0.721077	3.922080	4.173423	6.489578
L_θ	1.471040	4.241956	3.822231	6.815218
L_{ae1}	0.691187	3.138039	3.075141	4.478286
L_{ae2}	0.763051	3.413298	3.114875	4.908152
$L_{3/2}$	0.629899	2.944935	2.919861	4.223296

Table 6.5f: MCRE/10^{-2} to scale 12 or optimum for GS-AOC

Metric	Input	Gaussian	Impulsive	Mixed
Input	0	15.059435	9.564917	21.377353
L_1	2.088342	11.209543	4.306342	11.713694
L_2	2.042457	10.734013	4.314910	11.135845
L_Y	2.568110	13.671484	6.027975	14.989306
L_θ	1.815923	13.117531	4.057313	10.759898
L_{ae1}	1.874318	10.901930	4.155497	10.951389
L_{ae2}	1.849997	10.739651	4.092213	10.304398
$L_{3/2}$	2.050148	10.872554	4.298848	11.287057

the relatively low sensitivity of the MCRE to the vector magnitude, hence ignoring the handicap of the angular metric. The same happens with the luminance metric, yet for opposite reasons.

In terms of MCRE, the best one is the fist L_{ae} metric, which is as sensitive as the angular metric to

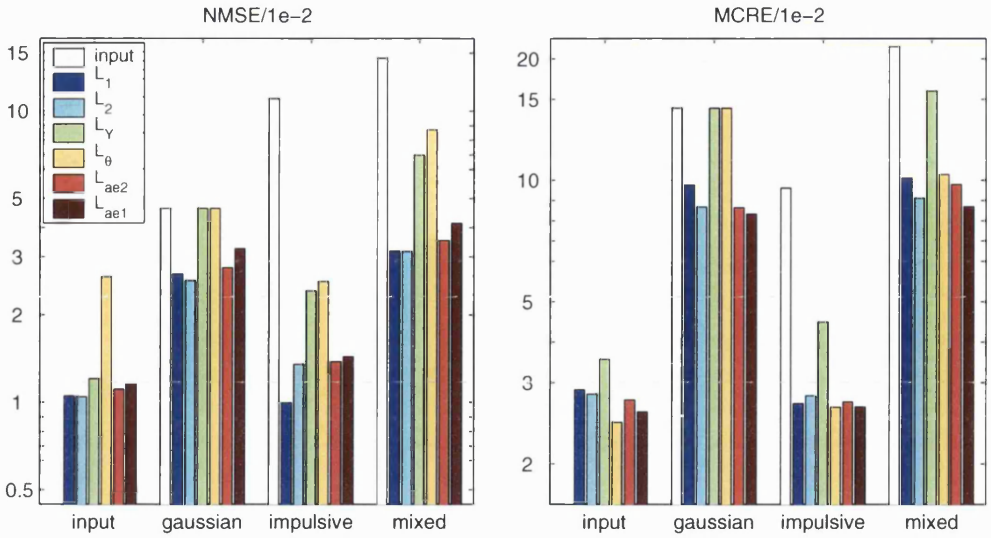


Figure 6.12: Metric vs noise, CCS

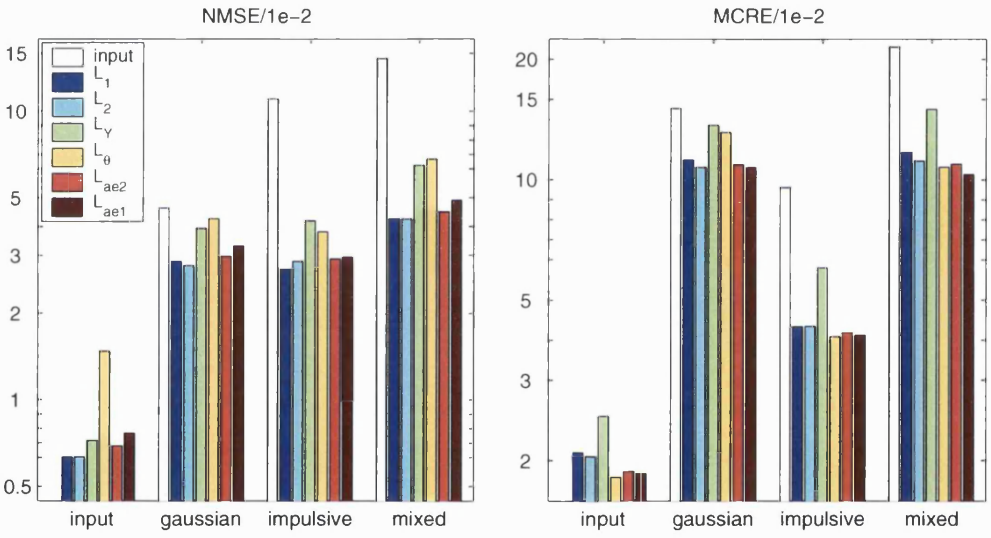


Figure 6.13: Metric vs noise, GS-AOC

chroma changes; all other metrics lag slightly behind even with increasing scales. In terms of NMSE, the best choice depends on the nature of the noise: for Gaussian-based, the Euclidean and the second L_{ae} give similar results, but when impulsive noise is present the city rule is a better choice. In addition, the city rule tend to produce better defined boundaries in these circumstances.

The reason for this effect is found in the metric distances. In essence, taking only greyscale values as neighbours, the nearest point to a black N-channel pixel corrupted in $\frac{N-1}{2}/\frac{N+1}{2}$ channels is the black/white point for the city metric, whereas for the Euclidean metric is the point with closest luminance

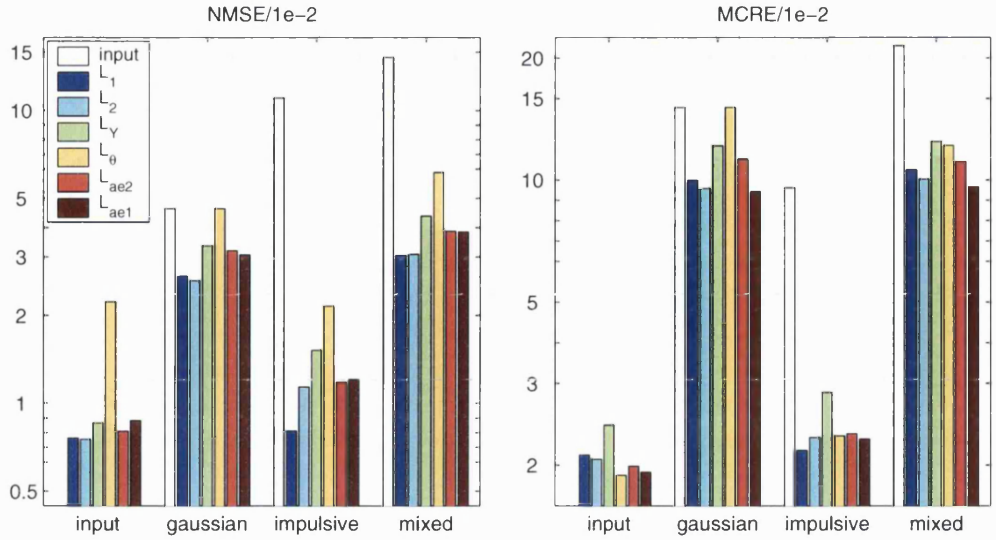


Figure 6.14: Metric vs noise, VAMS

value. Hence, the city rule is better at impulse removal near edges than the Euclidean and at keeping sharp edges, giving a sort of 'spatial coherence' from a visual sense. In conclusion, the best metric depends on whether impulsive or Gaussian noise dominates the image, and in the latter case, the importance of saturation and hue in the image itself, which favours the L_{ae2} metric.

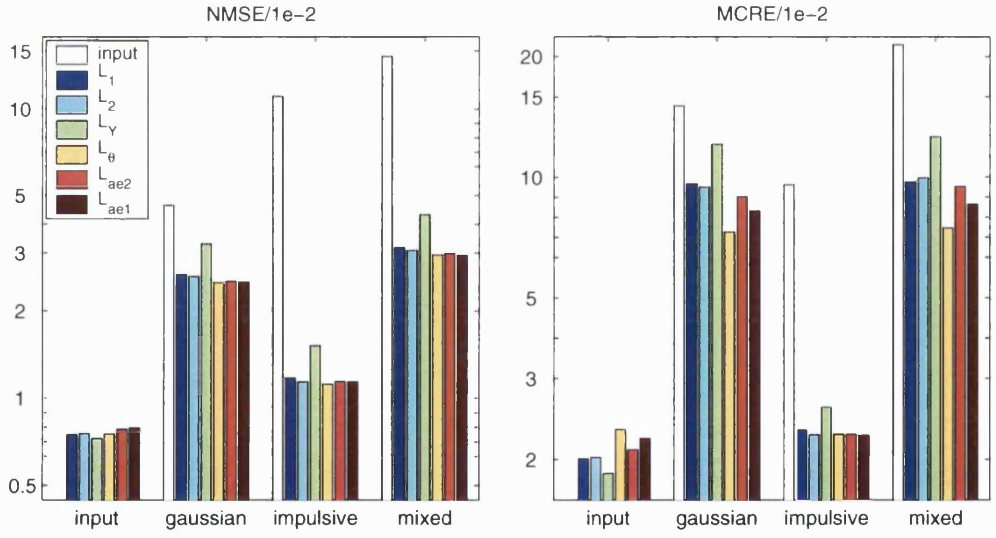


Figure 6.15: Gradient metric vs noise, VAMS

6.2.4 Extremeness metric evaluation

As mentioned before, the distance metric for the VAMS gradient affects the extrema definition and distribution. In all methods, the merging metric also affects noise removal and the proportion of outliers.

An evaluation was done, in terms of noise removal and extrema generation for the usual image subset; the merging metric for the VAMS gradient evaluation was Euclidean. For the extrema graphs, the scales considered were at fixed ratios (1%,2%,5%...) of the image size, being averaged amongst images.

Table 6.6a: NMSE/10^{-2} to scale 12 or optimum for the VAMS gradients

Metric	Input	Gaussian	Impulsive	Mixed
Input	0	4.624677	11.038768	15.205812
L_1	0.746918	2.661523	1.170602	3.297421
L_2	0.754737	2.620291	1.136560	3.229885
L_Y	0.726499	3.398570	1.512710	4.288777
L_θ	0.752319	2.495605	1.116581	3.115054
L_{ae1}	0.780548	2.522359	1.138777	3.145217
L_{ae2}	0.789837	2.510688	1.137826	3.115081
$L_{3/2}$	0.755791	2.637145	0.988960	3.198876

Table 6.6b: MCRE/10^{-2} to scale 12 or optimum for the VAMS gradients

Metric	Input	Gaussian	Impulsive	Mixed
Input	0	15.059435	9.564917	21.377353
L_1	2.002974	9.604409	2.358590	9.712210
L_2	2.021576	9.426907	2.295565	9.937106
L_Y	1.842883	12.05236	2.681658	12.57956
L_θ	2.366223	7.295148	2.301650	7.460138
L_{ae1}	2.107804	8.927684	2.298119	9.459562
L_{ae2}	2.248290	8.231904	2.288487	8.546515
$L_{3/2}$	2.075328	9.656846	2.274801	10.215253

Unlike in figure 6.14, figure 6.15 shows angular metrics provide the best performance and least distortion given a robust merging metric, with the luminance-based extrema detection faring the worst. The presence of uncorrelated Gaussian also has a greater effect on the colour information than on luminance, providing a more stable proportion of extrema for those metrics. Also, their actual amounts are slightly affected, initially favouring the luminance metric and to some extent L_{ae} metrics, although by minor amounts (only 100 more initial extrema for the 'lily' image).

Regarding the effects of merging metrics in figure 6.16 and 6.17, luminance and angle-related metrics again give slightly higher extrema counts in the VAMS, with little effects on the CCS and on greyscale sieving. That makes the angular metrics better at extrema detection; the initial advantage of luminance may be caused by misclassification of single channel differences.

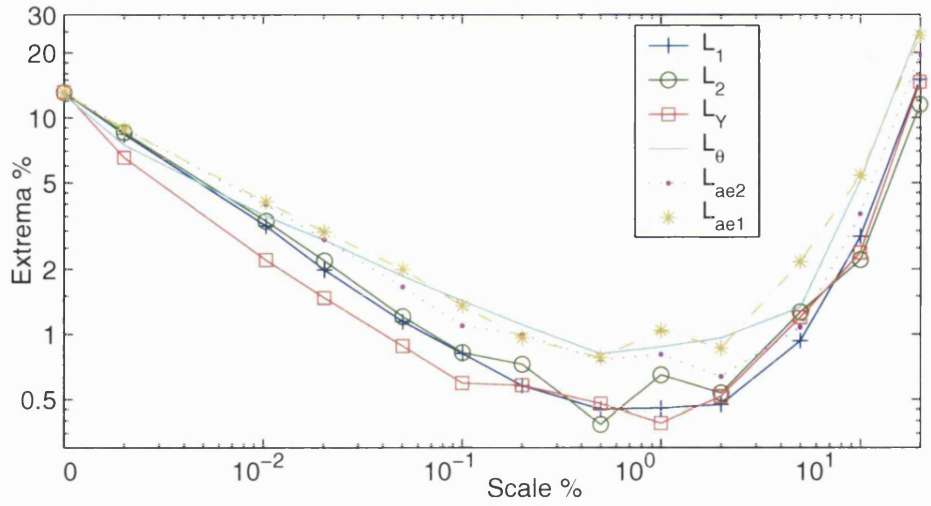
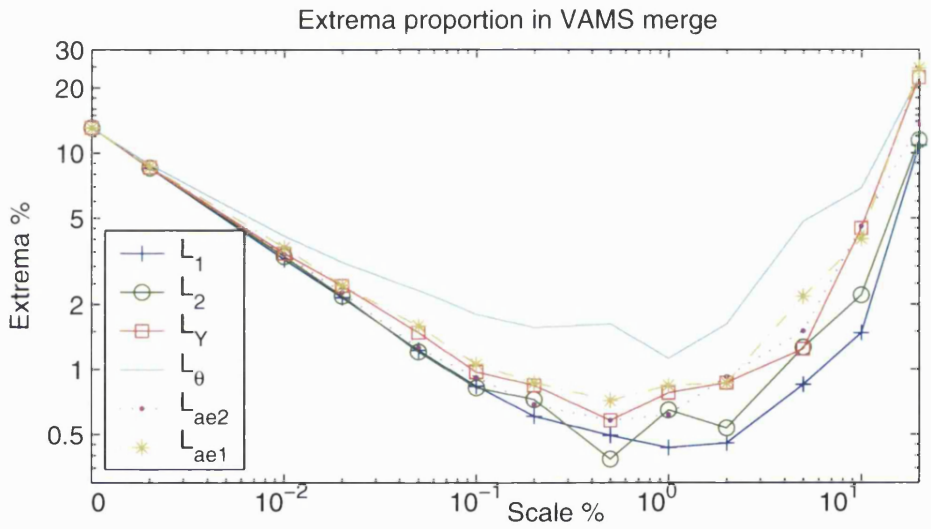


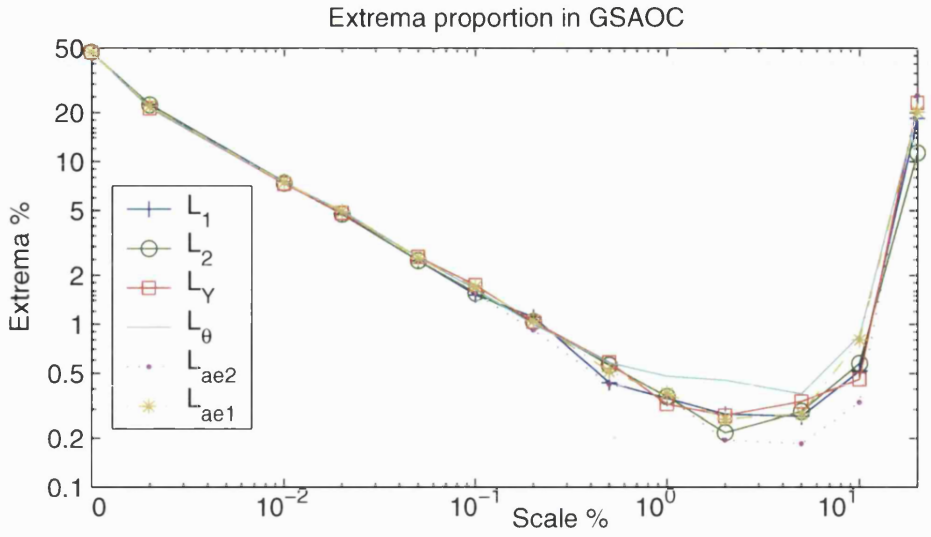
Figure 6.16: VAMS gradient metric



(a) VAMS merge metric, L_2 extrema

Table 6.7: Proportion of total initial extrema, VAMS

Metric	Input	Gaussian	Impulsive	Mixed
L_1	13.3670	13.1683	13.4010	12.8531
L_2	12.8487	13.1587	13.5972	12.8791
L_Y	13.4473	13.1937	13.5594	12.9515
L_θ	12.5881	12.8659	13.6239	12.7467
L_{ae1}	12.8151	13.2173	13.7204	13.0468
L_{ae2}	13.1196	13.2108	14.2403	13.3521



(b) GS-AOC merge metric, L_Y extrema

Figure 6.17: Extrema proportion vs scale

6.3 Centroid Merging

In morphology, the common approach is taking the value of the non-extreme region when merging, removing the extreme [56]. This means the noise performance of morphology is good against impulsive noise yet poor for Gaussian noise, as all values are altered and no valid references are possible. Pre-filtering to remove also noise improves performance in the segmentation.

Taking this idea, the choice of statistical values was used - taking the region centroid as its value rather than its original and merging to it. Two centroids were considered, mean and median. Since morphology implies no new values are created at any stage, two further options were taken:

- substitution; the centroid is taken as the new face colour value. This approach, although inducing new values, allows to carry out filtering at the same time as segmentation.
- guidance; the centroid is taken as the objective, and at merging the nearest vector is selected. No new values are created at any stage, proving more correct.

6.3.1 Noise Evaluation

Evaluation was done using the image set 'lily','boats','autumn','sample1', and L_{ae} metric. Noise conditions were Gaussian $\sigma^2 = 1000$, impulsive $\rho = 10\%$ and mixed, in each case uncorrelated. All options were evaluated: mean and median substitution (mean, med.), and mean and median guided (mean-g, med.-g) methods.

In clean conditions the mean gives 30% less NMSE and lower MCRE in any method, the median gives 20% reduction and guided method are midway, slightly worse than substitution. Generally the

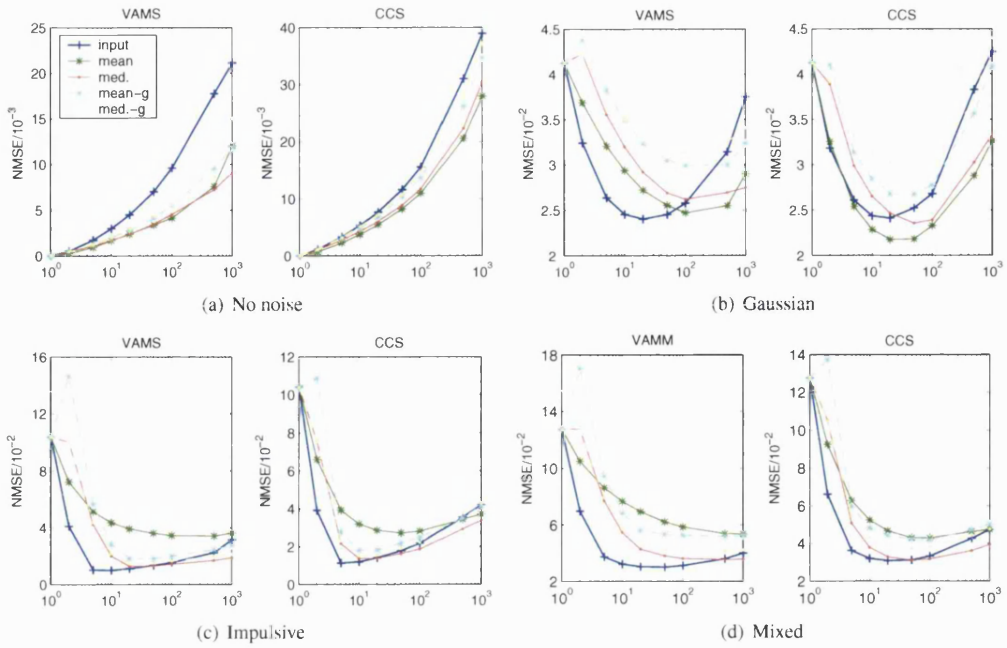


Figure 6.18: Merging selection: average NMSE

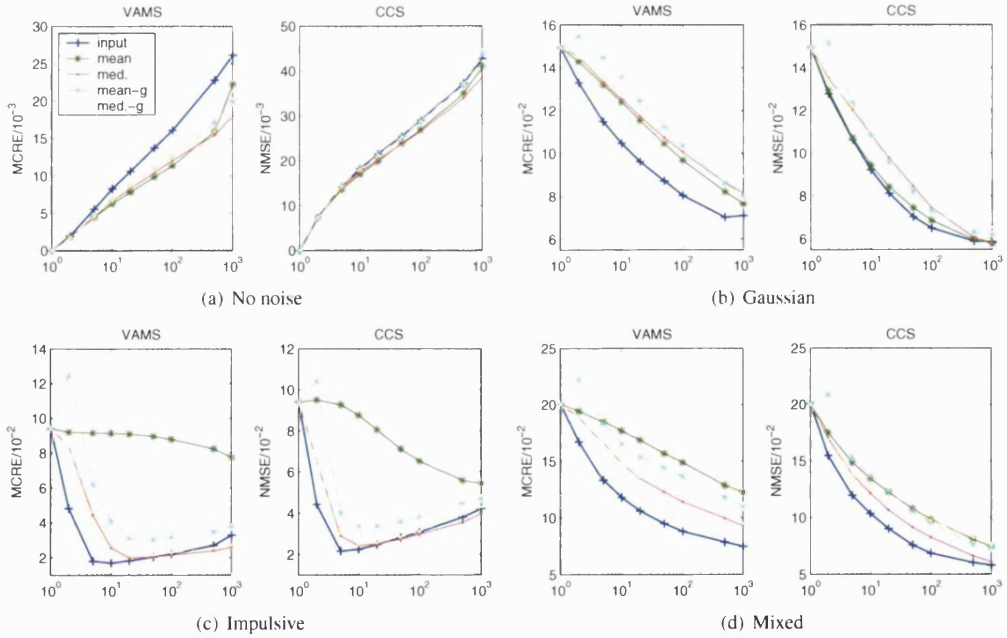


Figure 6.19: Merging selection: average MCRE

methods give less noise over broad ranges, although the difference may be minute (10% at best, 20% at much latter scales where no input colour is relevant); hence it is more suited for combined filtering and segmentation than for noise removal, as the minimum happens at scales too large compared to the original method. Results for the VAMS give better results for the centroid-mean 100 area sieve than for the original, but only for moderate Gaussian, as impulsive and mixed noise are bad for the mean and the median lags a bit behind, again a centroid-median 50 area sieve gives the best results. Chroma results are notably worse for the impulsive presence, and the guided methods do worse- the mean-guided is better against impulsive than the true one, but no match to the original. For the VAMOCS (see next section) things improve, performances keep the same but the NMSE stays the same for VAMOCS than VAMS, so more relative reduction for Gaussian. For the CCS, mean substitution gives a stable metric for large scales (low initial thresholds compensates for Gaussian, hence better) and 15% over standard merging; guided methods are 20% worse than the original like in the VAMOCS. Again, impulsive noise favours the median substitution as the mean guidance adds 20% more noise to the input. All metrics are similar at the lower end for all methods for the hull and VAMOCS in minima and evolution, but the VAMS goes better by being less aggressive and the hull overcomes the VAMOCS on quite heavy noise environs in the original, and surpasses both VAMS in all other variants.

For other images, results are similar: except for mean and Gaussian, all have a shallower curve (more distortion for clean images and less distortion at high scales elsewhere) and less robustness for guided methods. Thus, this can be used as a filtering device despite not being strictly mathematical morphology.

6.3.2 Discussion

Results illustrate the improvement of substitution variants over the guidance ones both in terms of NMSE and MCRE, except for the mean centroid when impulsive noise is present. This is caused by the region average being less representative for disperse data ranges, although at low scales this leads to worse metrics when the number of corrupted pixels excels the original ones. In all cases bar mean substitution on the CCS and noiseless conditions, the new variants achieve worse metrics and peak at higher scales, unencouraging their use. Other points to note are that calculation costs are higher for the guidance methods, and in all cases to the original method; also only the mean centroid substitution abides idempotency and causality in space (all others need the initial image) and since substitution variants are not allowed, that implies either breaking the rules of morphology or those of scale-space.

Results prove acceptable and interesting for the substitution methods in terms of quality, especially at higher scales where segmentation tones are more representative of the region itself. As such, it may have some applications to segmentation.

6.4 Conclusions

There are plenty of variations to the basic L_2 neighbour selection area sieves seen in the last section. Among the attributes chosen, contrast and power give the best results, improving over area for correlated or non-impulsive noise; for the metrics, the hue-sensitive L_{ae} metrics give similar results to the L_2, with

the usual tradeoff between chroma and luminance and further advantages in conjunction with the VAMS.

Although the basic parameters prove the most robust and good enough for most applications, the noise reduction properties and especially subjective appearance of colour sieves are definitely enhanced by the use of different attributes, metrics and even face values. In addition, features like oversimplification, seen among energy metrics in figure 6.11, are key to applications such as segmentation and simplification, as shown in the following chapter.

Chapter 7

Colour Image Segmentation

In prior chapters, evaluation has been noise-based and subjective in terms of segmentation and regions removed. Since one of the initial aims was using some psychovisual measure of goodness of region classification, the following gives the objective evaluation of their segmentation.

Segmentation is the process of grouping pixels into salient regions, as clusters related to actual objects or parts thereof, all enclosed by closed contours. This is a key stage in object-based systems (like MPEG4) and for automatic content-based retrieval, and to a minor degree compression, given a post-classification in foreground/ background regions. Linear scale-spaces produce a set of boundaries at any scale, giving a choice over the smoothing needed.

However, the graph morphology considered here produces a scale-space with strong causality, which means borders at any scale are a subset of previous ones. In addition, it easily adapts attributes besides (or together with) area and tree post-processing allow for results independent of sieving order, giving as many or more options to other region-based methods.

7.1 Measurement and methodology

Proper evaluation of any image processing method requires a large number of images and a consistent measure of performance, subjective or objective. This brings the problems of finding a database of images to segment, and a relevant metric to compare against.

Many authors, despite some common images readily available as in the USC-SIPI database [158], generally use their own database images and segmentations for results, without giving full details on their source or composition for easy comparison and benchmarking. This has gradually changed with more standard databases (and results) appearing [159], such as the MPEG-7 image dataset [160] used by Harvey et al. [145], the Corel databases, or the UCID (Uncompressed Colour Image Database) [161] as an uncompressed alternative to the MPEG dataset. The problem with all these sources is they are used and evaluated for content-based retrieval (a common use for segmentation) or are for general purposes, so segmentation of the image set is required. The Corel database also presents the problem of size, with

80 to 10000 images used by various authors.

A recent development is the Berkeley Database from Malik and Tal [17], based on a Corel subset. This publicly available dataset consists of 300 natural images (100 for testing, 200 for training), their segmentations by human subjects (an average of 5 per image for greyscale and colour versions) and the tools for obtaining precision-recall (P-R) curves for each image and overall; unlike the database in [145], these are already pre-segmented, besides being already tested for several graph and edge detection methods [75] [162].

Evaluating a set of segmentations often involves a number of test users, from which a Mean Opinion Score is derived; this subjective metric reflects the actual performance, on a 5-degree scale. The problem with this and other subjective metrics is the need for several independent evaluations per image, which can be problematic for multiple evaluations. Many segmentation metrics [163] are measures of goodness, revolving around the difference between the region-averaged segmentation and the original image, along with a penalty for over-segmentation or contour smoothness; their main problem is they measure desirable properties (smoothness, representative colours), without a reference to human results, and the measure can be coded into a method, making it perfect for that metric. Looking for more objective, discrepancy metrics with a reference to actual results, some options that were considered include the Global Consistency Error (GCE) and the related Local Consistency Error (LCE) [17], Pratt's Figure of Merit (FoM) [25] and precision-recall curves; also Zhang [163] proposes the minimum description length (MDL), as part of a co-evaluation framework. P-R curves were chosen, as they are part of the Berkeley Database and Malik's results give a quantitative reference to 'good' and random segmentations.

The GCE takes the normalised difference between matched border sets. Low or zero metrics appear when one segmentation can be obtained by simplifying another one, with the GCE being a more strict measure, i.e. $GCE \geq LCE$. Given a pixel p, contained by a region $R(S,p)$, contained in turn by a segmentation S_n with any number of regions, taking $A \backslash B$ as the set difference operator (belongs only to A) and with E as the pixel-wise local refinement error between segmentations,

$$E(S_1,S_2,p_i) = \frac{|R(S_1,p_i) \backslash R(S_2,p_i)|}{|R(S_1,p_i)|} \tag{7.1}$$

$$GCE(S_1,S_2) = \frac{1}{n} min\{\sum_i E(S_1,S_2,p_i), \sum_i E(S_2,S_1,p_i)\} \tag{7.2}$$

$$LCE(S_1,S_2) = \frac{1}{n} \sum_i min\{E(S_1,S_2,p_i), E(S_2,S_1,p_i)\} \tag{7.3}$$

Pratt's FoM is another popular method for edge detection performance. The definition is

$$FOM = \frac{1}{\max(S_D,S_I)} \sum_{i=1}^{S_D} \frac{1}{1+\alpha d_i^2} \tag{7.4}$$

where S_D and S_I are the actual and ideal segmentations, α is constant around 1/9 and d is the closest distance from the actual to any ideal pixel. The method is strongly dependent on the image and edge extraction method used. Pratt's figure is generally used with artificial images to evaluate edge detectors; however, Allen and Huntsberger give a colour segmentation evaluation [25] for region-based and edge-

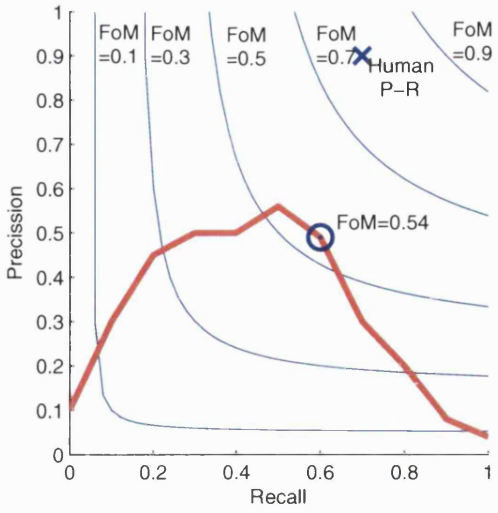


Figure 7.1: P-R example

based approaches based on it.

P-R curves are a variant of the ROC (Receiver operating characteristic) curves. ROC curves essentially plot false accept normalised by negatives (fallout) and true accept (recall) rates; in P-R, precision relates to true accept normalised by positives and recall is still the proportion of boundaries accepted, being independent of scale. In other words,

$$P = (truefoundpositive)/(foundpositive) \tag{7.5}$$

$$R = (truefoundpositive)/(truepositive) \tag{7.6}$$

$$F = (falsepositive)/(totalnegative) \tag{7.7}$$

$$FoM = (\frac{P^{-1}+R^{-1}}{2})^{-1} \tag{7.8}$$

As stated by Malik et al. [155], the precision-recall framework is closely related to the ROC curves used to evaluate boundary models, where the precision-recall curve captures the trade-off between accuracy and noise as the detector threshold is varied. Precision is the fraction of detections which are correct, while recall is the fraction of positives that are detected. This counters the fact that, as image size or resolution increases, the proportion of positives decreases since boundaries remain the same width (one pixel) and increase their length while the background increases in width and height, lowering fallout with resolution by a $\sqrt{N}$ factor.

Both metrics are computed using a distance tolerance below 1% of image dimensions to allow for small localization errors in both the machine and human boundary maps regardless of resolution. Unlike Pratt's FoM, that gives some tolerance to minor shifts and edge jitter, an important advantage given that different human subjects identify similar but not identical segments, giving some leeway to shifts.

P-R curves are also related to the GCE, taking a global match of a finite set of segmentations. Ho-

wever, they have the advantages over the GCE of not failing to trivial cases, allowing a fast scan of the complete space and being applicable to edge-based techniques; their limitation is the fact the GCE allows for an exhaustive search of the tree representation, something feasible but not desirable for P-R curves due to the loss of a threshold, together with timings [145] in either case. Other reasons for choosing P-R curves are the fact they are part of the Berkeley Database (although there are alternatives [164]), and its use by Malik and recently other authors of edge and region-based methods [165], allowing comparison. The example in figure 7.1 shows an example P-R curve with the point of peak FoM, together with contours of constant FoM; P-R among different subjects varies but is centered around FoM=0.80 .

From an analytical perspective, P-R curves are derived from the morphologically thinning of two segmentations, with true positives as points within 4.3 units (0.75% of the diagonal for 480*320 images) of a corresponding reference point. Precision and recall are derived from the normalised number of matches for each image, but for the entire database the cumulative number of matches and points for all images is used. Hence the overall P-R curve is not just the average of individual P-R curves, being biased in favour of the more complex images yet unbiased on a boundary basis. In all cases, evaluation was done on the 'test' set of images with region numbers; this allows the comparison of different attributes and a useful threshold that is more related to over- and under-segmentation than given attribute values dependent on the image.

7.2 Colour morphology segmentation

7.2.1 General terms

Some terms, used throughout this section, follow:

- undersegmentation/ oversimplification: from a reference image, the merging of most regions, removing salient features which are relevant and at or above the sieve threshold.
- oversegmentation/ undersimplification: from a reference image, the split of uniform regions into smaller, even pixel-sized, regions or the lack of simplification at some areas in the image.
- robustness/ stability: the effect of minor input alterations on the output; stable methods should give similar boundaries on corrupted images, whereas robust methods should give similarly good scores. They are not equivalent, although they are similar for methods with uniform simplification.
- aggressiveness: related to oversimplification, aggressive methods will remove most features below threshold or simply far more than another method.

7.2.2 General performance

All colour methods and the classic greyscale sieve (Y-AOC) were evaluated using the Euclidean metric for neighbour selection and 4-connectivity. The thresholds used were in region numbers, to give some independence from image contents and sieving attribute; also the Berkeley database definition encouraged subjects to split the image into a limited (2-30) number of segments. Region thresholds were at 9000,

5000, 1000 to 100 in steps of 100, 100 to 10 in steps of 10, 9 to 1 in steps of 2; that changed for some methods, giving more precision where required.

Method	Area		Contrast		Hull Contrast	
	FoM(R,P)	Regions	FoM(R,P)	Regions	FoM(R,P)	Regions
Y-AOC	0.37 (0.45,0.32)	3000	0.39(0.48,0.33)	3000	0.39(0.48,0.33)	3000
GSAOCS	0.47(0.48,0.45)	400	0.47(0.55,0.41)	1000	0.46(0.58,0.39)	600
VAMS	0.40(0.46,0.35)	60	0.35(0.42,0.30)	500	0.38(0.40,0.36)	70
VDMS	0.28(0.37,0.22)	1000	0.26(0.45,0.18)	10000	0.28(0.49,0.19)	5000
CCS	0.48(0.66,0.37)	30	0.50(0.65,0.41)	30	0.53(0.70,0.42)	30
VAMOCS	0.47(0.62,0.38)	40	0.50(0.66,0.40)	70	0.51(0.66,0.41)	50
VDMOCS	0.50(0.66,0.40)	70	0.52(0.65,0.43)	90	0.54(0.68,0.45)	70
PCS-V	0.47(0.62,0.38)	400	0.47(0.56,0.40)	500	0.50(0.64,0.31)	500
PCOCS-V	0.47(0.58,0.40)	20	0.50(0.68,0.39)	60	0.51(0.68,0.42)	40
PCS-A	0.48(0.66,0.37)	400	0.48(0.63,0.38)	500	0.50(0.65,0.41)	400
PCOCS-A	0.48(0.66,0.38)	40	0.50(0.68,0.40)	80	0.52(0.67,0.43)	50

Table 7.1: P-R FoM and ideal region threshold for main methods, 3 attributes on 100 images

(a) image (b) VAMS (c) VDMOCS (d) CCS

(e) human (f) GSAOCS (g) PCS-A (h) PCS-V

Figure 7.2: 'koala' example of best overall level segmentations

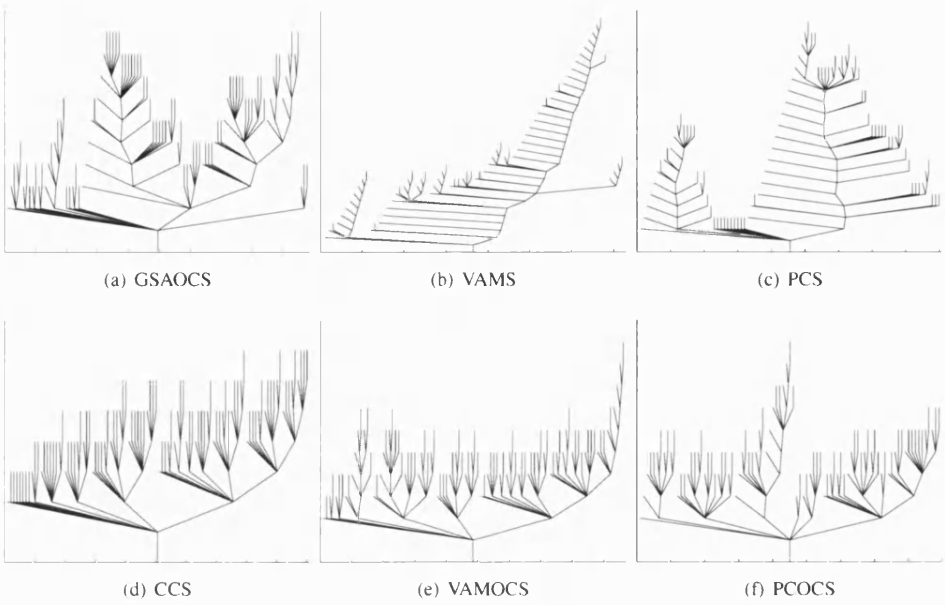
Note all results are for the RGB space and are very dependent on the processing space. As an example, input and Gaussian corrupted FoM results are 0.48, 0.44 for the VAMOCS and 0.49, 0.47 for the CCS; for Lab the VAMOCS gives 0.47, 0.42 and the CCS gives 0.48, 0.42. All other results for the database in table 7.1 and for one image in figure 7.2 show the VAMOCS and CCS are on par in segmentation performance; the GSAOCS, VAMS and VDMS are visibly behind and the VDMOCS is in the lead, with the PCS angular and vector sieves (PCS-A, PCS-V) slightly behind it and classic greyscale sieving further behind.

This can be explained by their different approaches to extremeness and edge information. For the particular example, the VDMOCS (FoM=0.666) and the CCS (FoM=0.662) give the most boundaries corresponding to strong edges, preferred by most humans subjects in figure 7.2e, while having less excess/missing features (respectively lowering precision/recall), with the VDMOCS oversegmenting most of the koala but getting most of the background plants, while the CCS oversimplifies the background plants on either side. The PCS-A (FoM=0.645) is behind, due to oversimplification of the lower half of the image with smaller objects yet it identifies many relevant boundaries (most of the tree and koala limits). The GSAOCS (FoM=0.605) and VAMS (FoM=0.600) are visibly worse because of excess regions at object transitions and the background respectively, while the PCS vector (FoM=0.590) is the worst in this case by a combination of all those factors.

In colour, extrema can be defined in more than one dimension as transitions or peak points, be it on the image gradient or the RGB space; both types need different treatment. The VAMS results in many extrema at edge regions, creating many unnecessary borders across uniform regions; the VAMOCS solves this by merging around minima, or representative regions, and the VDMOCS improves by being independent from area attributes. The CCS takes most points as extrema, being robust yet removing some fine detail before needed. As for greyscale sieves, they fail to take hue and saturation extrema although they will properly remove detail within visible objects (luminance-wise); the PCS openings find all these extrema types, giving a fair improvement, whereas the PCS M-sieves, by also removing edges, sets the boundaries independently of PCS maxima, countering their benefits.

Their tree structures in figure 7.3, and especially D.13 in the appendix, further show the differences in behaviour; here the leafs are the regions absorbed, where nodes with one leaf form vertical chains and nodes are horizontally ordered by their region number. Although most methods are similar, the VAMS (7.3b) clearly stands out; both the VAMS and especially the VDMS opening give an unbalanced tree with a dominant branch, resulting in a single object driving most of the sieve. The trees for the corresponding closings (see fig.D.13e-f) have a low height and many nodes off the root, reflecting a single region absorbing the others. This partly explains their poor segmentation performance and the major improvement of the VAMOCS and VDMOCS, with a number of branches off each node implying widespread merging.

For the PCS, openings are reasonably similar to the greyscale sieve in structure (a few seeds drive the sieve, giving a similar yet more balanced tree than the VAMS as seen in figure D.14), achieving their main objective; while combined with closing they give an even splitting of regions among branches as in the VAMOCS, with differences in performance stemming from the importance of size, shape and tone for each sieve as in figure 7.3a/c/e.



(a) GSAOCS (b) VAMS (c) PCS

(d) CCS (e) VAMOCS (f) PCOCS

Figure 7.3: Sieve comparison at 100 regions for 'lily'. See figure D.13 for more detail

The noise and segmentation performances also fit the links among methods in figure 7.4. The VAMS and its alternative, the VDMS, take the extremeness definition of the vector median, which gives good noise performance but is not ideal for simplifying smooth areas; considering minima gives the VAMOCS, with a good performance with either noise or segmentation. The CCS takes the idea of outliers by position as the greyscale sieve, but ignores the extremeness of its neighbours, being key to its aggressiveness with robust, but not optimum, results; all the PCS variants retake the VAMS idea of an extremeness field while assigning a score based on position and aiming for the GS-AOC behaviour (a few, stable extrema grow by absorbing transitions among them) while considering multiple channels, improving on its performance. Finally, the idea of minima sieving can also be applied to the PCS, giving a fast, widespread merging which counters the benefits of the GS-AOC

It should be noted that the evaluation method by Malik also includes a thinning stage, reducing the number of false positives from broad, thresholded gradient edges and improving results. Here the thinning process was kept, except when testing segmentation stability; in that case, matching all possible borders among similar segmentations becomes more important. The use of thinning affects lower scales, with many or all regions being unit area with adjacent boundaries; thinning these 'thick' boundaries gives partly random edges analogue to a watershed seeded from the larger regions. This is clearer in the GSAOCS, where untouched, broad region transitions (and adjacent boundaries) remain until very high scales and best results are at low thresholds. There, not thinning lowers the FoM by 0.05-0.15; all other methods quickly remove minor detail and therefore are unaffected.

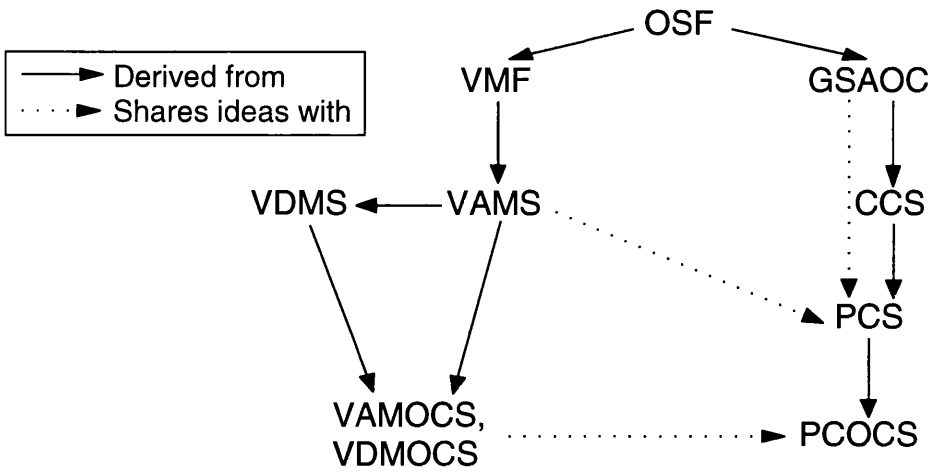


Figure 7.4: Relationship between methods

7.2.3 Noise robustness

The effects of noise on colour morphology in terms of its removal have been established; however its effects on segmentation performance need to be established. The main effects are rougher boundaries, lessened by 4-connectivity or the use of geometric constraints, and other changes at high scales as region vectors and saliency are altered. A basic evaluation is presented in table 7.2; this shows the effect of noise on detection of relevant regions, but not on the initial segmentation, i.e. the stability of boundaries.

The main procedure involved comparing the original segmentation, i.e. the optimum one, to that at different levels of noise; for example, the original segmentation may have a Figure-of-Merit of 0.40 yet its FoM against a corrupted boundary map may be 0.90, indicating a stable if rather inexact segmentation. Unlike in the previous sections, the boundary maps were not thinned, since the main effect of noise is new extrema and different boundaries when thinned at low scales, and the comparison is not to a thin human boundary map.

For simplicity and speed, a comparison was done between each level of the corrupted map and the uncorrupted one at the best level over a set of 40 images, with uncorrelated Gaussian noise of $\sigma^2 = 1000$, impulsive noise of $\rho = 1\%$ per channel and both combined; this visibly affects the average results, especially for corrupted images, but at said settings the FoM shift falls within 0.02 downwards, rising to around 0.05-0.10 for 20 images or less.

Results in table 7.3 are for the overall comparison rather than the average peak, explaining some of the low scores. As an example, the average best FoM score for the CCS with impulsive noise is 0.69 in the range of 50 to 300 regions left, whereas the best overall score is 0.60 at 100 regions left; this affects sieves more than edge methods, not having a single, ideal threshold for segmentation. An example is in figure 7.7; the original (blue) and corrupted (yellow) segmentations show the many edges induced by impulsive noise do not make the VDMOCS less robust to impulsive than to Gaussian, having more common (black) edges as shown by their respective FoM compared to the original.

Method	Input	$\sigma^2 = 100$	$\sigma^2 = 1000$
CCS	0.49(0.61,0.41)	0.47(0.60,0.39)	0.47(0.61,0.39)
VDMOCS	0.50(0.66,0.40)	0.49(0.60,0.42)	0.46(0.59,0.38)
VAMOCS	0.48(0.63,0.39)	047(0.58,0.40)	0.45(0.60,0.36)
PCS-V	0.48(0.65,0.38)	0.46(0.60,0.37)	0.45(0.65,0.34)
PCOCS-V	0.47(0.58,0.40)	0.47(0.59,0.39)	0.46(0.60,0.37)
PCS-A	0.48(0.65,0.38)	0.47(0.66,0.37)	0.46(0.59,0.38)
PCOCS-A	0.48(0.62,0.39)	0.48(0.65,0.38)	0.46 (0.62,0.36)
VAMS	0.40(0.51,0.32)	0.40(0.45,0.36)	0.40(0.61,0.29)

Table 7.2: Noise results for Gaussian corruption, main methods (FoM,recall,precision)

Method	Input	Gaussian		Impulsive		Mixed	
		to DB	to best	to DB	to best	to DB	to best
VAMS	0.377080 (0.49481, 0.30461)	0.378659 (0.65316, 0.26661)	0.552362 (0.44762, 0.72111)	0.371101 (0.48257, 0.30146)	0.449367 (0.39203, 0.52635)	0.394568 (0.56290, 0.30374)	0.46394 (0.34030, 0.72871)
CCS	0.475160 (0.55566, 0.41504)	**0.466221** (0.55085, 0.40413)	0.496353 (0.41258, 0.62282)	0.475008 (0.55621, 0.41449)	0.601339 (0.51127, 0.72992)	**0.454265** (0.54181, 0.39107)	0.491259 (0.40741, 0.61856)
GSAOCS	0.456985 (0.51156, 0.412929)	0.390044 (0.51752, 0.31296)	0.451709 (0.40417, 0.51192)	0.430742 (0.46804, 0.39895)	0.586155 (0.61297, 0.56159)	0.395539 (0.53026, 0.31541)	0.441955 (0.3608, 0.57024)
VAMOCS	0.465254 (0.62759, 0.36964)	0.434638 (0.60622, 0.33876)	0.481412 (0.36791, 0.69619)	0.462624 (0.68089, 0.35032)	0.552362 (0.44762, 0.72111)	0.437327 (0.61926, 0.33802)	0.489234 (0.38363, 0.67505)
VDMOCS	**0.479236** (0.61810, 0.39132)	0.451182 (0.58857, 0.36580)	0.487459 (0.42606, 0.56954)	**0.487531** (0.66661, 0.38429)	0.560078 (0.49108, 0.65164)	0.389455 (0.47732, 0.32891)	0.410186 (0.32415, 0.55841)
PCS-A	0.469217 (0.68489, 0.35684)	0.449048 (0.58598, 0.36399)	**0.546566** (0.43359, 0.73915)	0.457871 (0.66136, 0.35014)	**0.621264** (0.53156, 0.74738)	0.437671 (0.59817, 0.34508)	**0.540391** (0.42851, 0.73133)
PCOCS-A	0.465853 (0.65650, 0.36101)	0.455602 (0.63007, 0.35680)	0.50305 (0.41047, 0.64955)	0.478852 (0.66674, 0.37358)	0.574267 (0.47487, 0.72629)	0.451068 (0.62349, 0.35335)	0.501602 (0.41067, 0.64426)

Table 7.3: Robustness comparison: FoM(R,P) of input to human, corrupted input to human (DB) and to best level input segmentation for 3 noise models, area attribute

In all cases, Gaussian noise up to moderate amounts quickly worsens performance, with the CCS being the most resilient and the PCS the most stable, while VAMOCS and VDMOCS are more affected; this may be due to such methods being better suited for conservative noise removal (leaving other pixels unaffected) and not to highly changed numbers of extrema, altering the initial seeds and gradient field.

Further tests also compared all the altered map levels to those adjacent (i-1...i+1 at level i) in the original map. The resulting P-R curves for Gaussian corrupted images (σ^2=100 and 1000 in figures 7.5 and 7.6) follow a roughly straight line from the origin to (P=R=1) seen in figures 7.5b-c and 7.6b-c in contrast to the P-R curves to the initial optimum in 7.5d-e and 7.6d-e, with a different starting point for

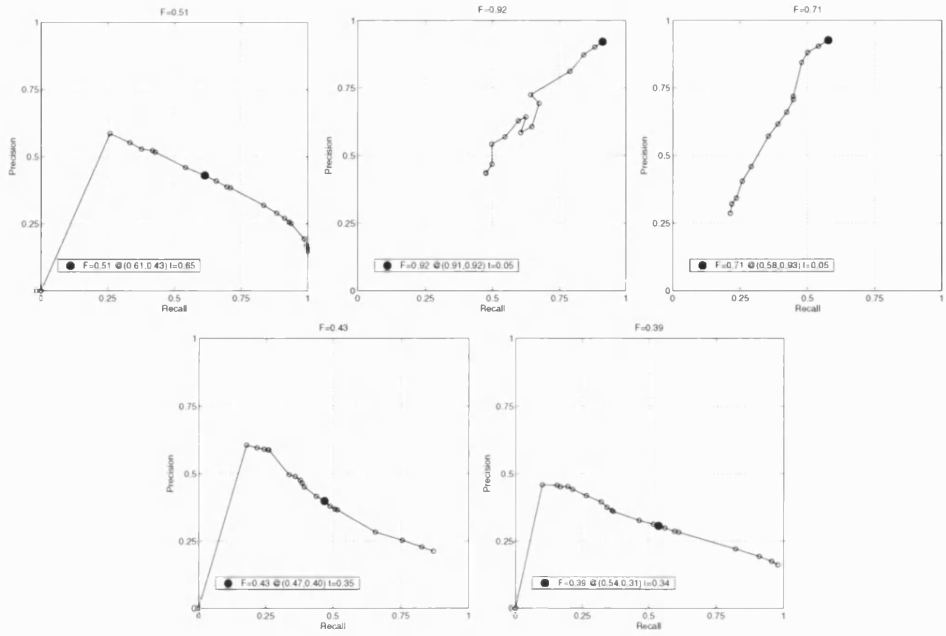


Figure 7.5: P-R curve of original GSAOCS, P-R of corrupted to method, and profiles to best match

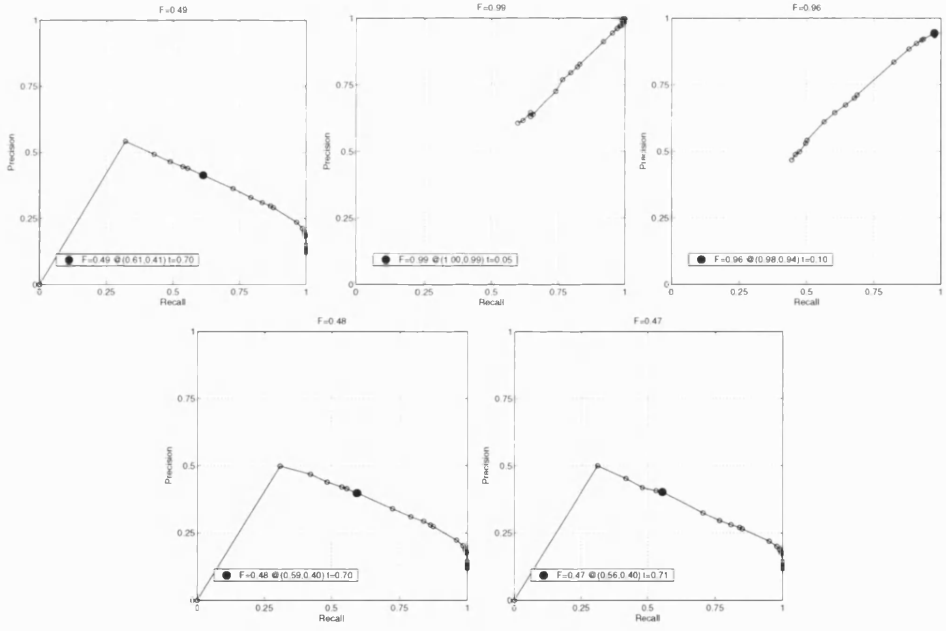
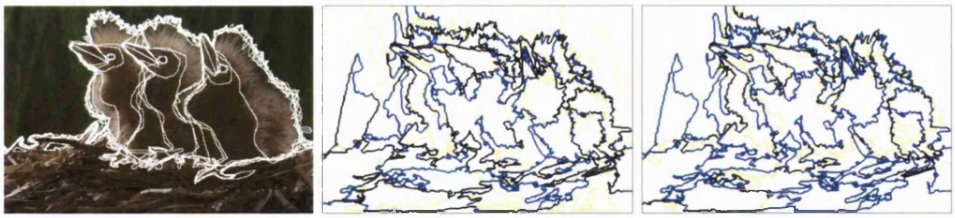


Figure 7.6: P-R curve of original CCS, P-R of corrupted to method, and profiles to best match



(a) Input (b) Impulsive corrupted (c) Gaussian corrupted

Figure 7.7: VDMOCS Border comparison: human (white) vs segmented (blue), closest impulsive /Gaussian corrupted (yellow) segmentations. FoM:0.49/0.56/0.48

each method for a given region number. The greyscale sieve curve begins much farther from (P=R=1) than the CCS, but this is an artifact due to region numbers; more importantly, it ends much closer to the origin, implying the final segments are less stable.

This determines the performance at the top level, region-wise, and could be taken as a measure, since all methods reach a figure of merit of 1 at the initial scale; however, the former method was used since the top level is rarely the best fit, and the latter is usually near the same level as in the original.

7.2.4 Attribute performance

Attribute segmentation was also evaluated. Only the CCS was considered in depth, dealing with all attributes and same conditions as table 7.3. Tables 7.4 and 7.5 give optimum and noise responses to all attributes, respectively. Table 7.1 gives a comparison among methods for the best attributes.

Method	Attribute	FoM(R,P)	Regions
CCS	Area	0.49(0.61,0.41)	30
CCS	Line Contrast	0.50(0.65,0.41)	30
CCS	Line Cont., full	0.51(0.67,0.41)	50
CCS	Hue Contrast	0.49(0.63,0.40)	50
CCS	Volume Contrast	0.53(0.70,0.42)	30
CCS	Volume Cont., full	0.54(0.70,0.44)	30
CCS	Entropy	0.48(0.63,0.39)	30
CCS	Inertia	0.48(0.62,0.39)	30
CCS	Power, to value	0.46(0.59,0.38)	30
CCS	Volume, to centroid	0.48(0.63,0.38)	50
CCS	Volume, to value	0.47(0.64,0.38)	50
CCS	Area & Contrast	0.50(0.66,0.40)	50
CCS	Volume & Cont.	0.48(0.65,0.39)	50
CCS	Cont. & Vol.Cont.	0.48(0.65,0.39)	50
VDMOCS	Area	0.51(0.61,0.43)	70
VDMOCS	Area & Contrast	0.52(0.62,0.45)	70
VDMOCS	Area, Cont. & Vol.Cont.	0.54(0.66,0.45)	90

Table 7.4: FoM results for CCS, all attributes

Regarding attributes, the best ones are those contrast-related and any which is fairly independent of area; this was expected, since the main drawback of an area constraint is the split of uniform regions.

Attribute	Input	Gaussian		Impulsive		Mixed	
		to DB	to best	to DB	to best	to DB	to best
Area	0.4752 (0.5557, 0.4150)	0.4662 (0.5508, 0.4041)	0.4886 (0.3751, 0.7005)	0.4750 (0.5562, 0.4145)	0.6013 (0.5113, 0.7299)	0.4543 (0.5418, 0.3911)	0.4913 (0.4074, 0.6186)
Contrast	0.4903 (0.6858, 0.3815)	**0.4847** (0.6247, 0.3960)	**0.5845** (0.5149, 0.6757)	**0.4957** (0.6792, 0.3903)	**0.6087** (0.5215, 0.7311)	0.4695 (0.6570, 0.3653)	0.5171 (0.4503, 0.6072)
Volume	0.4520 (0.6683, 0.3414)	0.4598 (0.5252, 0.4088)	0.4910 (0.4201, 0.5906)	0.4568 (0.5134, 0.4114)	0.6076 (0.5519, 0.6758)	0.4457 (0.5161, 0.3922)	0.4825 (0.3914, 0.6291)
Power	0.4714 (0.5544, 0.4099)	0.4625 (0.540, 0.404)	0.4798 (0.3829, 0.6424)	0.4704 (0.5441, 0.4142)	0.5326 (0.5607, 0.5071)	0.4673 (0.5464, 0.4082)	0.4937 (0.3889, 0.6758)
Hue Cont.	**0.5180** (0.7031, 0.4101)	0.4808 (0.5359, 0.4360)	0.5327 (0.4361, 0.6843)	0.4906 (0.7123, 0.3741)	0.604 (0.5468, 0.6745)	**0.4796** (0.5408, 0.4308)	0.5208 (0.4517, 0.6147)
Volume Cont.	0.5020 (0.5149, 0.4898)	0.4604 (0.5478, 0.3971)	0.4834 (0.4056, 0.5982)	0.4906 (0.7123, 0.3742)	0.6039 (0.5467, 0.6745)	0.4774 (0.5283, 0.4355)	**0.5213** (0.4535, 0.6130)

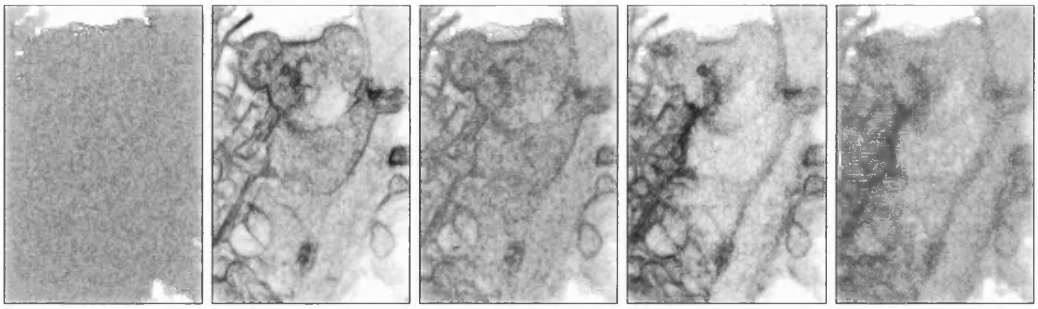
Table 7.5: Attribute comparison: FoM(P,R) of input to human, corrupted input to human (DB) and to best ideal segmentation for 3 noise models, CCS method

Those attributes also are the most resilient against low levels of noise, probably for the same reasons. In terms of noise removal, however, the area related attributes are better as seen on table 7.5, despite giving always lower results as in table 7.4. Such differences can be seen through increasing scales in the sieve tree, as shown in figure D.12 for the greyscale sieve. Although similar at high scales, the area tree gives a more spread structure at the initial scale and the greatest change with scale regarding the nodes removed off the main branches, simplifying all small features and noise. Contrast gives a tree with a few predominant branches at all scales, pointing at a stable segmentation, whereas power falls in between these two cases.

The results also favour multi-attribute region minimization sieves with a slight edge over classic attributes, which may be due to the overlapping results of the attributes studied, see figure 6.5. A better measure for multiple attributes might be to minimize not the number of regions potentially removed, but the area or the energy present i.e. the proportion of the image likely to be affected. For single attributes, line contrast and hull volume give the best results whereas power is the worst attribute; reinforcing the idea that area-independent attributes fare best when it comes to segmentation.

As a side note, using a random segmentation gives an insight on the influence of edges and features on an attribute independently of the method or distance metric. Shown in figure 7.8 are some sieve results for 50 iterations of different attributes: area shows no relation, followed by volume, power, contrast and hull volume in terms of stable (darker) boundaries, which happen to overlap strong gradient edges. Since gradient edges are often object boundaries (see figure 7.2e), that gives some incentive to using contrast, in particular the external or full versions, over other attributes, especially when the method or metric is not the best suited for that image or needs to be robust (e.g. the PCS or CCS), as also reflected in table

7.1 for contrast and area.



(a) Area, Volume (b) Full Contrast (c) Power (d) Full Hull Cont. (e) Internal Hull Cont.

Figure 7.8: Effect of attribute on random segmentations

7.2.5 Discussion

Different methods have different segmentation performance. Area is the most robust attribute against noise, but those less related to area give the best results. In a similar pattern, methods related to the greyscale definition of extrema prove more robust against noise yet worse for pre-processed or ideal images.

In all cases, the measures of performance and precision seem rather low if compared with edge methods already evaluated by Malik et al. (a peak and average FoM of 0.65(R=0.69,P=0.61) and 0.60, compared to 0.54(R=0.68,P=0.54) and 0.48), especially for the VAMS. This is not unexpected for a number of reasons.

First, edge methods provide unconnected border pixels near high gradient locations (high precision, low recall) whereas region methods give connected boundaries, often within objects (low precision, high recall). Removing a weak boundary implies merging any adjacent regions, and in turn segmenting a smooth transition implies introducing other less relevant boundaries; that gives a low precision unless the image is oversimplified, where any extra merging can quickly lower recall.

Besides, important boundaries are often related to strong edges. Edge methods remove weak borders first hence giving better precision and performance with little effort; region methods do so by merging internal regions. With mathematical morphology this can only be done up to a global attribute threshold, either removing relevant small regions or giving over-segmentation at any one level. Morphological methods also work on their prior output, altering the saliency of regions at increasing scales.

Finally, the random segmentation provided by the database follows the same trend as related methods. Random edge segmentations consist of random magnitude pixels, giving scattered edge points when thresholded. For region methods, the equivalent is area sieving of random extrema with random best-fit selection, giving highly irregular boundaries for sieves. Since using connected boundaries increases recall while lowering precision even further, using a random segmentation leads to worse figures of merit; using random margins and 100% of extrema gives F=0.38, compared to F=0.43 from Malik.

7.3 Watershed segmentation

A classic method in image processing, the watershed's main purpose is segmentation, with few if any uses in noise removal. The watershed uses the image gradient, flooding from the minima upwards and placing boundaries where different flooding basins meet.

A key problem of this technique is over-segmentation due to noise/ texture. This can be improved by using the waterfall/ recursive watershed, or (as examined here) filtering the image or gradient. The main options in colour are a vector gradient [140] and the maximum trimmed gradient (taking the maximum distance, removing the points involved as noise and taking the distance again [166]), with moderate gains.

The VAMS gradient gives a new alternative. This method involves using a metric-based quantity to create an image in which extrema are found. Such an image can be used as the input to the watershed, like a classic gradient image. Such a watershed scheme has similar problems to the original (over-segmentation); however, sieving the image (by VAMS) or the gradient itself solves this. Removing all minima below a threshold attribute does give a better segmentation, both on the original image or on the gradient image itself; good attributes are contrast (like with openings by reconstruction), power (for high enough values) and volume (as an improvement to power). Note that sieving before taking the gradient simplifies some of the topology, but post-gradient operations have a more beneficial effect.

See appendix C for results, and figures D.9, D.8 for some watershed examples. Shafarenko and Petrou [43] give a similar operation, using openings/closings by reconstruction on the gradient in an iterative approach.

7.3.1 Results

As expected, AM pre-processing of the watershed gradient improved the overall results. In particular, the use of attributes on the gradient gave the best effects.

On the watershed of the processed luminance, as expected from table C.2, Area morphology gave more relevant results than SE morphology while removing far more detail, keeping more relevant boundaries. Mean and especially median filtering are in turn better at simplification, keeping lower region counts for a given MSE and lower mask sizes. Hence the median and area AOC are the best, altering mainly peak flat zones. The main problem with either method in colour or greyscale, though, is they give a number of flat zones with clear borders, eventually blocking watershed growth.

However, the use of gradient sieving gave the best segmentation results, even exceeding colour sieves - but still exceeded by many edge techniques. Compared to the prior methods, the VAMS gradient fields gives much greater simplification for a given distortion, between half and a third of the median, making it an useful alternative to greyscale gradients. VDMS closings are clearly worse mainly due to their widespread oversimplification, yet there is little difference between both variants when including openings. Finally, from the basic methods studied the median is clearly ahead; the truncated median is worse in this case, with its edge sharpening abilities being counterproductive.

For all other greyscale sieves and except for contrast, removal of all extrema is outperformed by closings; as seen in figure C.1, many of the borders absent from, or weaker in, closings are fine texture not

considered by the test subjects in figure C.1c, boosting precision. This can be explained by the absence of gradient spikes in the image set, with the benefits of smoothing in the PCS and GSAOCS opposed by broader edge plateaus in most cases; although the external contrast was used, affecting regions with sharp edges, it only accounts for a small part of this gap (from FoM=0.429 to FoM=0.440 for external and internal greyscale contrast closings). Besides contrast, all attributes perform equally well regardless of method, with volume and power being better with greyscale and colour data respectively. This suggests that the main factor in watersheds is the attribute, not the use of colour.

7.4 Tree pruning

An alternative option to sieving for the trees produced is tree pruning [129] [74] [125] . Tree-based and region-based AM can give different results: the reason is the tree pruning step; and as noted by Garrido and Salembier [56], the simplification operator is also possible. Also the region-based AM stops when a region is no longer a maximum, whereas pruning stops when the region has reached the attribute; they can differ since they are not linked in colour and extremeness is not a stable measure.

Given the trees produced, these sieving options are available:

1. to a branch level
2. to an attribute scale, increasing or not
3. to a set number of branches/regions
4. any of the above, removing all possible branches or only those with removed predecessors

Two of the advantages of tree sieving are the 'simplification' [56] operator, or removing regions other than the extrema, and more ease to use multiple attributes. The simplification operator is best suited for area, although it can be used to some extent with not strictly increasing attributes.

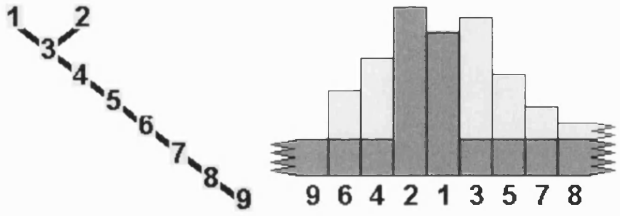
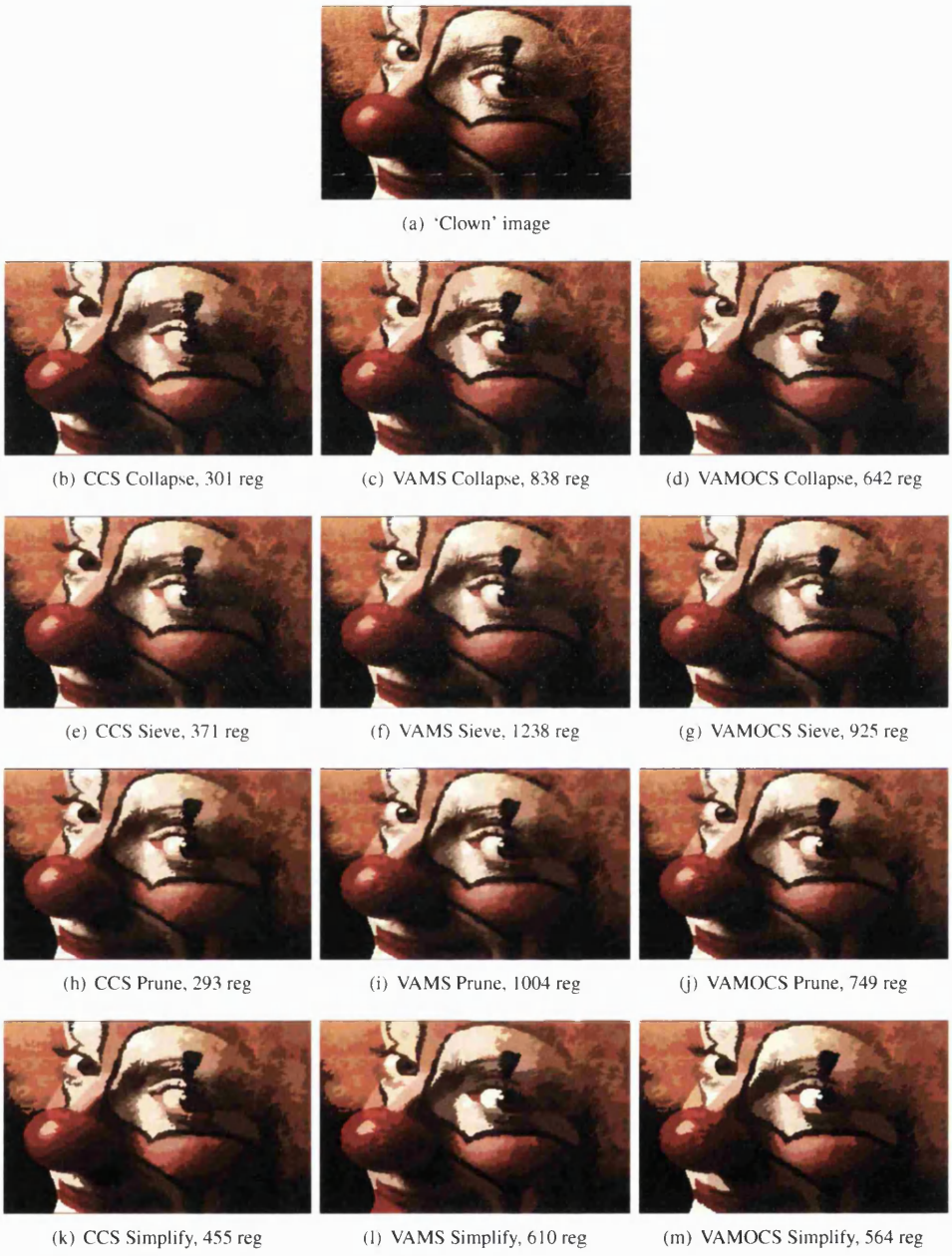


Figure 7.9: Tree collapse problem: region before/after, and tree

The main problems of tree processing are scale-space causality and connectivity, which helps the former. Only by removing child nodes within each parent node in the order of insertion gives always a proper segmentation (i.e. those physically closer and in contact are removed), and only when removing all nodes within a branch can it be assumed that all nodes are one connected set. Also, the sieve can only be reproduced by removing nodes in the overall sequence, since the merge-dependent extremeness status after each merge is missing from the tree. This gives a set of split regions when collapsing, giving

a potentially different tree from that created as seen in figure 7.9 above. There, taking closings-openings and considering regions 8,9 as much larger, collapsing to the top node will give one (1) rather than two (1,2) leaf nodes. This does not happen with either openings or closings, coincidentally.



(a) 'Clown' image

(b) CCS Collapse, 301 reg (c) VAMS Collapse, 838 reg (d) VAMOCS Collapse, 642 reg

(e) CCS Sieve, 371 reg (f) VAMS Sieve, 1238 reg (g) VAMOCS Sieve, 925 reg

(h) CCS Prune, 293 reg (i) VAMS Prune, 1004 reg (j) VAMOCS Prune, 749 reg

(k) CCS Simplify, 455 reg (l) VAMS Simplify, 610 reg (m) VAMOCS Simplify, 564 reg

Figure 7.10: Example of simplification, pruning :area CCS and VAMOCS sieves, at area 25

All data has been obtained by sieving to idempotency; there is, however, the option to sieve to other thresholds if idempotency is not the main condition [67]. In the area sieve examples of figure 7.10,

all cases of simplification/ collapsing (first and last rows) have the erased regions merged to the leaf nodes for clarity and pruning involves the removal of all leaf nodes. Note pruning refers to removing all feasible extrema below threshold, while simplification removes all leaf nodes below threshold. As seen, simplification (the sieve of all leaf regions below threshold, last row in fig.7.10) removes the most regions for the VAMS-based methods and gives larger uniform patches (removing the detail at the eye and cheek present in fig.7.10b-c), but collapsing proves a better overall segmentation in all methods since pruning gives few changes in all three methods. The only problem is the colour artifacts in the VAMOCS and VAMS (see the pink region on the forehead in fig.7.10b-c, being white in the sieves in fig.7.10e-f) caused by extrema generation, which lets large, minimum regions grow by taking maximum (leafs), swapping the right values.

Comparing the trees among methods gives that GSAOCS gives a high depth tree which is fairly unbalanced and branches off to the root. The VAMS and PCS keep some of those properties while giving a more balanced tree, while combined open-closings give a fairly balanced tree. The CCS gives a tree close to binary in structure, with multiple branches coming off the parent nodes. In summary, the proportion of extrema and the production of new ones determine the structure of the tree. One possible processing option is stable salient contours, as in [67], although the face value should be based on a statistic value rather than that of the middle or last node, which can cause colour artifacts.

7.4.1 Results

The methods implemented are prior forms of sieving (sieving to region or attribute threshold), together with simplification and a variant of the stable salient contours (SSC) [67].

Looking at figure 7.10, using simplification removes all regions below threshold; this shows little change when used with the CCS, underlining its aggressiveness and that relevant regions are not tied to a single attribute or level. In contrast, the other methods, especially greyscale and VAMS (fig. 7.10l), suffer extensive pruning due to the presence of long chains of small regions where the top leaf was not one of the starting outliers.

The use of stable Salient Contours (SSC), a form of tree collapsing suited for segmentation and reconstruction, was the next step in research; collapsing is done until the threshold is exceeded or a second branch under collapse is met, whichever goes first. This differs from more usual definitions, where variance/ histogram thresholding of the nodes determine the path length to be removed.

Once again, processing the CCS tree shows little change, unlike the other methods. Collapsing the VAMS (fig. 7.10b) gives more, larger, flat zones than the VAMOCS (fig. 7.10c), mainly because the VAMOCS tree has more extrema and therefore shorter branches before they meet each other, hence less pruning; also minima often are large regions (the background, the shadows and any uniform regions in the VAMS image) or surrounded by maxima (like the initial texture), which is also the reason why the VAMOCS sieve looks identical to the VAMS in this image.

Overall, collapsing has a greater effect on the VAMS and GSAOCS methods than on the other methods explored; in the CCS differences are often few, if any. The overall P-R measure only improves slightly in those two methods and the PCS-A; the main cause appears to be the absence of long chains

Method	Input	Collapse	Simplify	Prune
VAMS	0.39690	0.39927	0.41690	0.34639
	(0.43826,0.36268)	(0.47361,0.34510)	(0.50590,0.35453)	(0.69338,0.23086)
VAMOCS	0.48175	0.47064	0.46886	0.47408
	(0.62654,0.39132)	(0.62021,0.37920)	(0.61872,0.37744)	(0.60601,0.38932)
GSAOCS	0.46424	0.55456	0.40614	0.40722
	(0.51130,0.42510)	(0.75812,0.43717)	(0.66253,0.29283)	(0.49862,0.34413)
PCS-V	0.47576	0.47797	0.43955	0.44369
	(0.66382,0.37074)	(0.69810,0.36338)	(0.57146,0.35712)	(0.74641,0.31567)
PCOCS-V	0.48182	0.47835	0.44483	0.45787
	(0.66335,0.37830)	(0.66645,0.37306)	(0.64907,0.33836)	(0.58414,0.37648)
CCS	0.49080	0.47434	0.47776	0.47916
	(0.55518,0.43980)	(0.55475,0.41429)	(0.61596,0.39021)	(0.6610,0.37588)

Table 7.6: Best level results for original, collapsed methods: FoM(P,R)

of nodes in most colour morphology methods compared to greyscale in figure D.13, giving few potential improvements. However, greyscale sieves can provide longer branches (see appendix D.14a), so the minor improvement in segmentation of the VAMS seems appropriate. The use of pruning, being similar to simplification, has few positive effects.

One last option is decomposing the tree by step changes when collapsing, keeping nodes at regular attribute intervals. Using other attributes besides area, then sieving and pruning essentially modifies the lower ranks or leaves, without any further effects.

7.5 Conclusions

Morphological segmentation is a complex area. Colour gives much more choice as shown in the last chapter, and interesting applications such as new watershed gradients, but it is advanced tree processing which gives the most options and potential. All of the methods, common to region-based techniques ignoring texture, have difficulty with heavily textured images in the database, especially those of animals with clear patterns on colour-wise similar backgrounds or uneven lighting, while doing well wherever there is a clear contrast among objects. In such cases, including texture along with colour components when defining a region [53] might solve the problem.

Among the methods studied, the worst is consistently the VDMS, followed by the VAMS and luminance sieve. However, the methods derived from them and that rely on edge information, namely the VDMOCS and to a lesser extent the VAMOCS, give the best results on smooth images and unsurprisingly are less robust to noise (or the cases described above), in which case the CCS and PCS give the most relevant and reliable segmentations respectively. The PCS also gives the best noise removal, which probably explains its stability.

Changing attributes gives similar effects in performance to changing attributes; here, the best one is the hull contrast, while the external Euclidean line contrast is more robust and has almost identical scores, followed by all their other variants. This comes to show the relevance of gradient information, and the limits and possibilities of standard and colour morphology.

Chapter 8

Conclusions

This thesis has studied the application of advanced techniques to colour images. Despite the recent work in colour imaging, most morphological techniques still rely on a greyscale conversion before application. Here, new morphological scale-spaces that are more suited to colour and their applications to segmentation and noise removal were studied.

The lack of efficient colour sieves has been a key drive in the development of this research. Chapter 1 is an introduction to image processing and the importance of colour in vision, image corruption and segmentation methods. As shown, there are potential uses for new colour techniques in the area of displays. Basic noise removal and segmentation techniques based on gradients and clustering have been easily extended to colour; however, region- and morphology-based processing are still largely unexplored, driving the thesis into that field.

A review of nonlinear image processing methods came in chapter 2, giving common greyscale shortcomings and the more popular nonlinear methods, median filtering. Their extensions to colour were reviewed and evaluated, showing their advantages over their greyscale counterparts. Also, a correlated model of noise for multichannel images was introduced here.

Chapter 3 examined another family of nonlinear methods, greyscale mathematical morphology. These operations have some desirable properties, and their main operators and extensions to colour in classic literature were described. Their shortcomings, especially the definition and removal of extrema, were the motivation for further research into colour and sieves, along the lines of Salembier et al. [125] by splitting the concepts of extremeness, metric closeness and region removal which are linked in greyscale.

Two recent colour morphological sieves were examined in chapter 4. These methods, the VAMS and CCS, share their sieve algorithm. Their differences and advantages over colour and area morphology are shown. More importantly, they give better noise rejection than greyscale, pointing at better extrema definitions and merging options.

The superior results of the evaluation in chapter 4 motivated further research into sieves in chapter 5. As their main differences in the evaluated methods stem from extremeness definitions, new colour sieve structures were proposed. Including the VAMS gradient minima to its extrema list gave the VAMOCS;

extending the CCS definition from a discrete decision gave the PCS, and by extension the PCOCS. An evaluation of all these methods considering their structure, timings and robustness against noise was undertaken. Considering noise removal, they offer similar performance to median filters and better resilience against Gaussian noise than greyscale morphology extensions, other than marginal ordering; in this method, the creation of new hues still gives the best noise metrics.

Other features of colour sieves were expanded in chapter 6. Different attributes were exposed and extended to colour, including entropy, surface and volume as new colour attributes, together with other non-increasing criterions more suitable for thinnings. As shown in the case of contrast, their definition often proves unclear, even unpractical, in higher dimensions. New distance metrics were also considered, especially energy metrics. Finally, the use of statistics in the region values was evaluated, including median or mean filtering in the sieve; their results proved better at medium and high scales except against impulsive noise, which could be of use when extracting large objects.

Chapter 7 finally discussed the overall work and possible applications to the area of segmentation. A quantitative segmentation performance evaluation and other forms of tree postprocessing were also considered with the Berkeley database [17], a subset of the Corel database already used by other researchers [53] [162]. Results show that edge-related sieves, despite less robustness, achieve better results. This fact and their successful application to watershed preprocessing point to the importance of gradient for vision.

There are several areas for further work. Segmentation has also become more popular recently [53] [167] so applications of this research include it, obviously. Taking this, novel morphology methods, and improvement of other methods (e.g. watershed pre- and post-processed scale-spaces) are the main options.

PCS improvements, as the better colour extension of the GSAOC, and tree processing at different levels/attributes, or considering proximity features are also good areas for further work, not to mention colour transforms among other changes. Tree processing is especially important; by forgoing idempotency, and in some cases either merging order or classic connectivity, it is feasible to obtain an optimum segmentation, or a set thereof, from the sieve tree. Sieves can then be considered as tree preprocessing; the colour sieves described already give better segmentation scores than their greyscale counterparts regardless of the tree operation, which points to a more relevant set of segmentations and a better starting point.

Energy metrics and more gradient related sieves are also promising, judging from the basic metrics implemented here and their boundary smoothing properties. Galun et al. [53] give another direction for possible developments, taking cumulative measurements; again, this conflicts with face value changes and semi-group structure, or getting similar results from any earlier scale besides the input.

Applications of all the sieving techniques shown includes improvement of other methods, such as watershed pre- and post-processed scale-spaces. Image enhancement and noise removal by using image sieving to low scales is possible; here, the high speed of the VAMS-based methods at low scales or small images may allow for almost real-time processing if done in parallel. Finally, there is automatic image segmentation, possible by using attributes dependent on the image contents (e.g. power, contrast, volume) or on the desired output (e.g. power, simplicity, entropy).

Appendix A

Author's Publications

D. Gimenez and A.N. Evans. Colour Morphological Scale-Spaces for Image Segmentation. In *Proc. British Machine Vision Conf.*, Oxford, volume 2, pages 909–918, September 2005.

D. Gimenez and A.N. Evans. Colour Morphological Scale-spaces from the Positional Colour Sieve. In *Proc. Digital Image Computing: Techniques and Applications*, Cairns, pages 415–422, December 2005.

D. Gimenez and A.N. Evans. An Evaluation of Area Morphology Scale-Spaces for Colour Images. In *Computer Vision and Image Understanding*, Accepted Manuscript, http://www.sciencedirect.com/-science/journal/10773142. Springer-Verlag, March 2007.

Colour Morphological Scale-spaces from the Positional Colour Sieve

David Gimenez and Adrian N. Evans
Dept of Electronic and Electrical Engineering
University of Bath
Bath, BA2 7AY
United Kingdom
{eepdg , A.N.Evans}@bath.ac.uk

Abstract

Morphological sieves are a popular tool for scale-space image analysis. Recent work has considered the development of colour and multichannel sieves, and their application to image segmentation. This paper proposes a new colour sieve based on the geometry of the local convex hull, providing a more flexible approach to extrema definition. Results show the new approach to have a similar segmentation performance to existing colour sieves but with an improved performance in terms of noise reduction.

1. Introduction

Scale-spaces formed from greyscale images are an important tool for hierarchical image analysis. Traditionally, the scale-spaces are linear and are created by employing the diffusion equation [11]. More recently, the application of a series of structuring elements of increasing scale have been used by several researchers to form morphological scale-spaces, for example see [10] and [14]. However, in common with linear scale-spaces, scale-spaces based on structural morphology do not obey the property of strong causality with the result that new boundaries can be created, and the position of edges can drift, with increasing scale [1]. Alternatively, morphological sieves based on connected operators possess the property of strong causality and, as they employ area operators, require no a priori knowledge of the shape of image objects [2, 1]. In addition, sieves have low computational complexity and are robust to noise [9].

Morphological sieves are useful for many image processing applications. At small scales they can be used to remove image noise [19, 18]. At larger scales, sieves produce regions that show correspondence with image objects and have been formally related to segmentation algorithms based on region merging/classification [6]. Consequently, they have been employed in applications such as segmentation and classification [16, 1]. Although colour plays an important role in the segmentation process [13], the development of multichannel sieves is problematic as their underlying morphological operations require regional maxima and minima to be identified and processed. As no unique ordering for multivariate data exists, these operations cannot easily be extended to colour images.

The convex colour sieve (CCS) [7] and the vector area morphology sieve (VAMS) [5] were proposed independently in 2003 and address the problem of extending morphological sieves to colour images. These two approaches, although differing in the details, have algorithms that essentially follow the same steps and the main difference in their performance results from their approaches to defining extrema [8]. The CCS forms a convex hull from each pixel and its connected neighbours and then defines a pixel as extreme (resp. non-extreme) if it lies on the edge (resp. interior) of the hull. The VAMS first forms a scalar image in which the value of each pixel is the mean aggregate distance to its connected neighbours, assessed using a norm, and then identifies extrema as the maxima in the scalar surface. Recent analysis has shown that the binary decision used by the CCS produces a very high proportion of extrema and an aggressive filtering action. In contrast the VAMS has fewer extrema and hence a lower computational cost. However, a lower proportion of extrema also results in less image simplification for a given scale. The VAM Open-Close Sieve (VAMOCS) increases the aggressiveness of the VAMS by also sieving the minima in the scalar surface produced by the VAMS [8]. These minima correspond to "nearly flat" regions and their inclusion in the merging process was shown to produce an improved segmentation performance.

As the surface produced by the CCS is binary every pixel is either extreme or non-extreme and the inclusion of minima would result in every pixel participating in the merging process. This paper proposes a new sieve, termed the Positional Colour Sieve (PCS), which addresses this prob-

lem by deriving a scalar surface from the geometry of the local convex hull. This process reduces the proportion of pixels that are categorised as extreme and also allows the closing operation of the VAMOCS to be incorporated into the PCS structure, giving the Positional Colour Open-Close Sieve (PCOCS).

This paper is organised as follows: Section 2 reviews colour sieves and describes the PCS algorithm. A performance evaluation is presented in section 3, including noise reduction and a quantitative evaluation of the segmentation performance using the Berkeley segmentation dataset. Finally, conclusions are given in section 4.

2. Colour sieves

A main steps of a generalised colour sieve algorithm are given in [8] as:

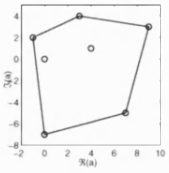
1. Identify all extreme regions;
2. Merge all scale 1 extrema regions with their nearest neighbour;
3. Repeat steps 1 and 2 until no extrema are found at current scale;
4. Repeat steps 1 to 3 with increasing scale until only 1 region remains.

Compared with its better known greyscale counterpart that processes the maxima and minima separately, colour sieves simply process extrema as they cannot distinguish between maxima and minima. The merging process in step 2 is analogous to that of the greyscale sieve and changes the colour of each extreme region to that of its closest neighbour, as assessed using the Euclidean distance. However, the merging is not guaranteed to produce a non-extreme region and, in addition, can create new extrema close to the merged regions. As a consequence, in colour sieves the addition of step 3 is necessary to ensure idempotence. Steps 1 and 2 are therefore repeated at each scale and, as the number of extrema found depends on the definition of extrema adopted, this is a critical factor in the performance of the sieve.

The CCS defines extremeness in terms of membership of its local convex hull: if a point is on the edge of the local convex hull consisting of the point and its connected neighbours, then it is extreme. This method reduces to a combined opening and closing for greyscale images and can be extended to any number of dimensions. In addition, its extremum definition is also invariant to rotation and linear transformations of the axes. The main disadvantage of this method is the number of extrema: a n-dimensional hull needs at least $n+1$ points to be non-degenerate, which means a high proportion of the points are classified as extreme, giving an aggressive sieving action. For example, figures 1(a) and (b) show a complex image and the convex

0+0i	3+4i	3+4i	2−3i	4−5i
4+1i	4+1i	3+4i	0−7i	3+2i
9+3i	0+0i	0+0i	−1+2i	3+2i
9+3i	7−5i	7−5i	−1+2i	2+1i
0+2i	0+2i	7−5i	−1+2i	2+1i

(a) Complex image



(b) convex hull for $0 + 0i$ region

17.00	29.00	29.00	32.50	18.00
27.00	27.00	29.00	**81.00**	26.00
39.00	38.00	38.00	43.00	26.00
39.00	**66.67**	**66.67**	43.00	11.00
38.00	38.00	**66.67**	43.00	11.00

(c) VAMS scalar image using L_1 norm

5.6009	4.8543	4.8543	0.0040	3.4123
3.1419	3.1419	4.8543	**5.0654**	4.5705
5.0680	0.1045	0.1045	4.3594	4.5705
5.0680	5.0877	5.0877	4.3594	3.3012
5.3871	**5.3871**	5.0877	4.3594	3.3012

(d) PCS scalar image

Figure 1. Colour sieves example using 8nn connectivity. Extrema marked in bold in (c) and (d).

hull for the $0 + 0i$ region. As $0 + 0i$ lies inside the hull it is not extreme. However, using this definition, there is only one other non-extreme region in the image. As well as increasing the processing load, classifying the majority of pixels as extreme does not fit well with an intuitive interpretation of extrema as outliers.

An alternative definition of extrema is provided by the VAMS. Here, extrema are the maxima of a scalar surface in which each region is assigned the average aggregate distance from each pixel in the region to its neighbours. In figure 1(c) this results in only 2 extreme regions, whose values are clearly different from their neighbours. Minima in the scalar image correspond to regions that are closer to their neighbours than their connected neighbours are to theirs and the VAMOCS also identifies and processes these, producing an improved segmentation performance [8]. In figure 1(c) the 4 corners are minimum regions, with similar values to their surrounding points.

The PCS aims to derive a scalar surface from the geometry of the local convex hull. This approach has several advantages: unlike the VAMS, the extrema will not be affected by rotation of the axes, the number of extrema will be more in line with expectations, and it will also enable minima to be identified and processed. In the approach adopted here, the value of each point in the scalar image depends on the following criteria:

1. The angle with neighbouring points. For points on the exterior of the hull the angle is measured using

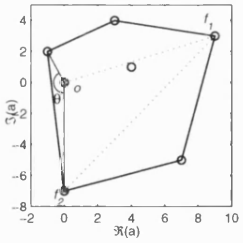


Figure 2. PCS example for 0+0i region from figure 1(a).

neighbouring exterior points. For points inside the hull the maximum angle to neighbouring exterior points is used. When all points are in a plane the maximum angle is π so, to avoid bias, all angles are scaled to a $0 - \pi$ range. As the angle reduces the point becomes more extreme, so it is subtracted from the maximum angle to give an increasing measure.

2. The normalised distance, measuring the extremeness of the position. The first step in finding this measure is to define 3 points, the original point (o), the point furthest from o (f_1) and the point furthest from f_1 (f_2). The normalised distance is then given by $(of_1 + of_2)/f_1f_2$ or $2of_1/f_1f_2 = 2$ when o and f_2 are the same point. The measure is shifted to a range of 0–1 by subtracting 1.

A measure of extremeness is then given by the product of the angle and the normalised distances measures. This measure overcomes the problem of using the angular measure in isolation, as points that are close to coplanar can be differentiated by the distance measure. Similarly, points that are located amid similar neighbours are less likely to be classified as extreme. Finally, a third criterion is added to ensure that points within the hull are not classified as extreme:

3. Whether the point is within or on the edge of the hull. As extrema are not expected to occur within the hull, the scalar product of measures 1 and 2 is augmented by π for points on the edge of the hull. Points coplanar or collinear with edges are considered to be in the hull.

This process is illustrated for the $(0 + 0i)$ region previously considered (see figure 2):

- The maximum angle is $\arccos(-14/\sqrt{5 * 49}) = 2.678$ rad (using points $(0-7i)$ and $(-1+2i)$), giving a measure of $\pi - 2.678 = 0.464$ rad.

scale	VAMS	CCS	GSAOCS	PCS	PCOCS
1	2209	33616	4062	8889	17433
	41207	41207	41207	41207	41207
2	1054	12852	1973	3602	5695
	37199	16402	36553	30775	22189
10	151	2211	424	667	134
	27547	2878	28457	15548	5020
50	21	456	110	918	189
	19660	572	20697	7546	1132
100	8	223	63	71	96
	16102	279	17040	5435	559
500	1	44	11	13	19
	8024	53	9114	2498	97
1000	1	23	7	7	12
	6871	25	6150	1478	40

Table 1. (Number of extreme regions)/(total number of regions) for Lily image.

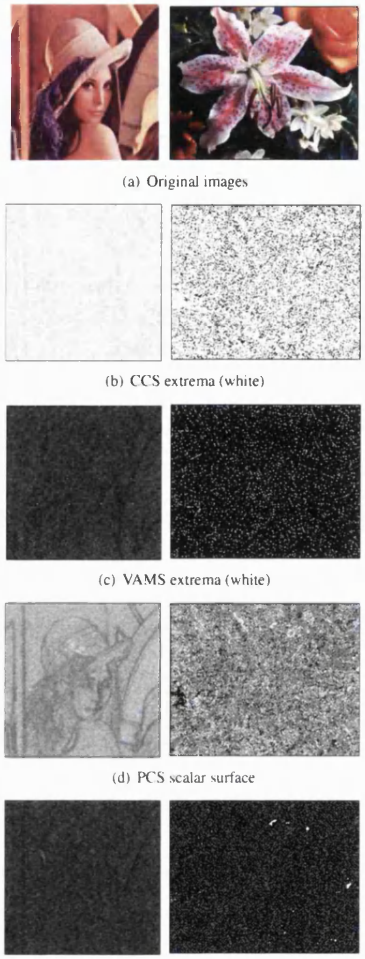
- Point $f_1 = (9 + 3i)$ and is 9.49 from point o (0+0i), and point $f_2 = (0-7i)$ is 13.45 from f_1 and 7.00 from point o.

- Since $f_1f_2 \neq of_1$, points o and f_2 are not the same there is no need to scale d_2. The normalised distance $(9.49 + 7)/13.45 - 1 = 0.225$

- The final measure is $0.464 \times 0.225 = 0.1045$.

Repeating the above process for all regions in figure 1(a) produces the PCS scalar image shown in figure 1(d) in which there are 3 extrema (maxima) and also 3 minima.

Figure 3 shows the distribution of initial extrema at selected scales for the test images Lily and Lenna. The fraction of Lily's regions that are extreme are given in table 1 and were similar to the fractions obtained for Lenna. The PCS initially has roughly four times the number of extreme regions as the VAMS but only a quarter the number of the CCS; this factor of four occurs by chance. As each extremum has to be merged with its closest neighbour, the aggressiveness of the PCS's sieving action should therefore lie somewhere between the VAMS and the CCS. The fractions for the PCOCS in table 1 also includes minima, which relate to nearly flat regions. The PCOCS also processes the minima regions in the scalar surface and, as scale increases the total number of regions reduces through the merging process; this explains why the PCOCS achieves a more rapid reduction in total regions, and hence more image simplification.



(a) Original images

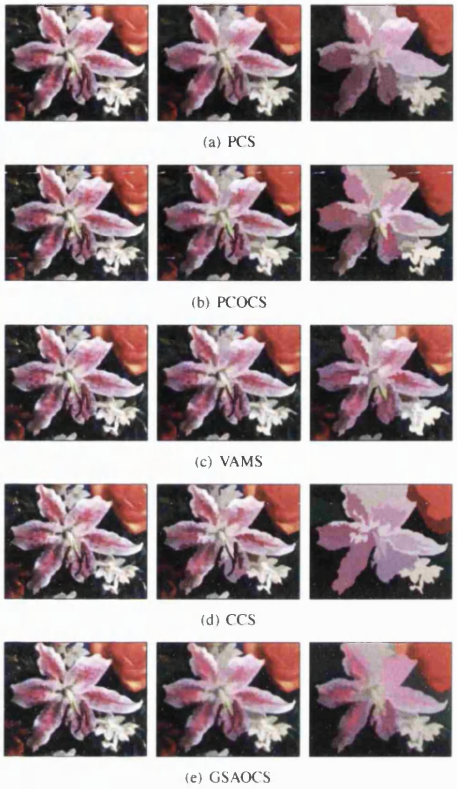
(b) CCS extrema (white)

(c) VAMS extrema (white)

(d) PCS scalar surface

(e) PCS maxima

Figure 3. Colour sieve extrema for Lily and Lenna images.



(a) PCS

(b) PCOCS

(c) VAMS

(d) CCS

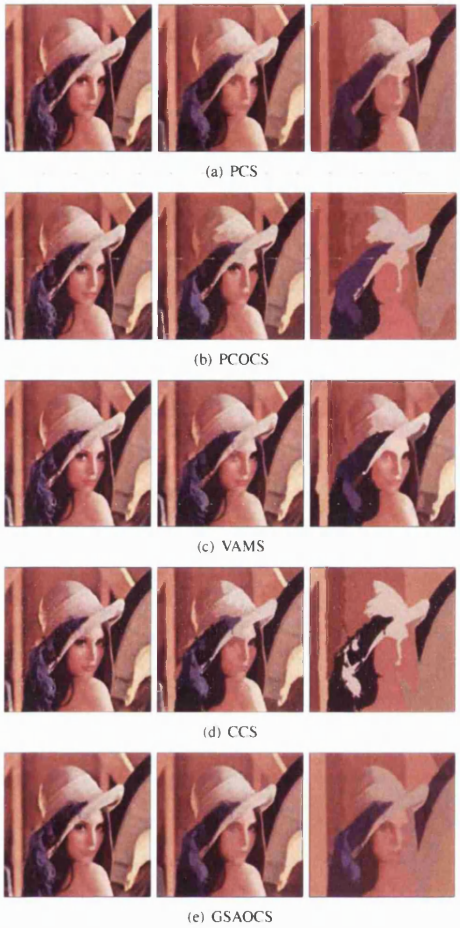
(e) GSAOCS

Figure 4. Sieve results for Lily image. Area = 10 (left), 100 (centre) and 1000 (right).

3. Experimental results

The PCS was evaluated on a number of natural images and the results for the Lily (186×230 pixels) and Lenna (512×512 pixels) are representative of those obtained. For comparison, the images were also sieved with the CCS, the VAMS and the VAMOCS. All colour sieves used 4 nearest neighbour connectivity and the L_2 norm. In addition, results for the greyscale area-open-close sieve (GSAOCS) were obtained, using brightness as the ordering factor.

The results of sieving the Lily and Lenna images are presented in figures 4 and 5. All sieves preserve significant boundaries through scale. An indication of the aggressiveness of the sieving action is given by the amount of image simplification at each scale and the results show broad agreement with the proportion of extrema detailed in the previous section; those sieves that have the highest proportion of extrema produce the most image simplification at a given scale. Figures 4 and 5 show the aggressiveness of the



(a) PCS

(b) PCOCS

(c) VAMS

(d) CCS

(e) GSAOCS

Figure 5. Sieve results for Lenna image. Area = 100 (left), 1000 (centre) and 10000 (right).

PCS falls between the VAMS and CCS and a quantitative evaluation of the segmentation performance is presented in section 3.2.

3.1. Noise Reduction

To evaluate the ability of the sieves to remove noise, the Lily test image was corrupted with light (1%) and moderate (10%) impulsive noise and with moderate ($\sigma^2 = 10^3$) and heavy ($\sigma^2 = 10^4$) Gaussian noise. In colour image processing, the normalised mean square error (NMSE) and the mean chromaticity error (MCRE) are two widely used metrics that provide objective error measures [17]. The NMSE is given by

$$NMSE(\mathbf{f}, \overline{\mathbf{f}}) = \frac{\sum_{x=1}^{M}\sum_{y=1}^{N} \|\mathbf{f}_{xy} - \overline{\mathbf{f}}_{xy}\|^2}{\sum_{x=1}^{M}\sum_{y=1}^{N} \|\mathbf{f}_{xy}\|^2} \quad (1)$$

where M and N are the image dimensions, and $\mathbf{f}_{xy}$ and $\overline{\mathbf{f}}_{xy}$ are the original and processed pixels at location (x, y) respectively. The MCRE is defined as the distance between the intersection points of f_{xy} and $\overline{f}_{xy}$ with the Maxwell triangle. When the triangle is defined on the unit plane this gives

$$MCRE(\mathbf{f}, \overline{\mathbf{f}}) = \left(\sum_{x=1}^{M}\sum_{y=1}^{N} ((r_{xy} - \overline{r}_{xy})^2 + (g_{xy} - \overline{g}_{xy})^2 + (b_{xy} - \overline{b}_{xy})^2)^{1/2} \right) / (MN) \quad (2)$$

where r, g and b are the normalised RGB values [3].

The noise-corrupted images were sieved over range of areas and, at each scale, the NMSE found. The minimum NMSE and corresponding MCRE for each sieve are given in tables 2 and 3 respectively. Results show that with no added noise the GSAOCS produces the least image distortion. When noise is added, the PCS has the lowest NMSE and MCRE measures for all noise types except light impulsive noise, where its performance is similar to the GSAOCS.

Method	-	Imp1	Imp2	Gaus1	Gaus2
-	0	1.17	10.77	4.56	26.13
VAMS	0.83	0.38	1.40	3.18	9.57
CCS	1.15	0.49	1.64	3.10	9.69
GSAOCS	0.40	0.21	1.80	3.45	17.95
PCS	0.52	0.25	1.19	2.44	8.52

Table 2. NMSE ($\times 10^{-2}$) for colour sieves applied to Lily image.

Method	-	Imp1	Imp2	Gaus1	Gaus2
-	0	1.05	9.95	17.56	35.12
VAMS	1.55	0.66	2.07	13.01	19.56
CCS	2.57	1.30	2.59	11.83	16.06
GSAOCS	1.03	0.49	2.49	15.01	30.30
PCS	1.64	0.66	1.86	10.20	16.75

Table 3. MCRE ($\times 10^{-2}$) for colour sieves applied to Lily image.

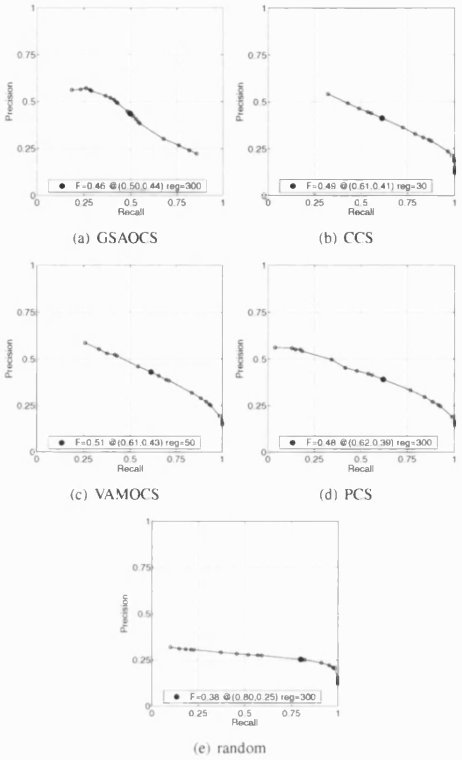
This shows that the definition of extrema used by the PCS successfully identifies noise points. It was also found that for higher levels of noise the minimum NMSE occurred at larger scales for lower noise levels. The PCOCS was not included in this evaluation as the main benefit of its use of closings is for segmentation rather than noise reduction.

3.2 Image Segmentation

An evaluation of the segmentation performance of the colour sieves was undertaken using the Berkeley database [12], a set of natural images and human ground truth segmentations. To provide a quantitative performance measure precision-recall (P-R) curves of [13] were employed. These are a variant of ROC curves where precision is the normalised number of identified boundary pixels (true positive/total positive) and recall is the proportion of identified pixels or the false-negative rate (true positive/total true positive). These metrics have the advantage of being independent of scale, unlike the ROC curve, and are an alternative to other metrics such as the global coherency error (GCE). They also enable segmentation performance to be characterised by a single figure (the F-measure), given by the harmonic mean of precision and recall.

By increasing the scale until the sieve images contained a fixed number of regions, and varying the number of regions, P-R curves for each sieve were obtained. Figure 6 presents the average P-R curves for 100 images from the dataset with the corresponding F-measures given in table 4. Results show the segmentation performances of the CCS, PCS and PCOCS to be very similar. Although the processing of minima by the PCOCS reduces the number of regions at which the peak F-measure occurs from 300 to 30, the F-measure is unchanged. This is unlike the VAMOCS where, in comparison with the VAMS, an improved F-measure occurs with a reduced number of regions. For comparison, the curve for a random segmentation is also shown. This was generated by marking all regions as extreme and randomly selecting the neighbour to merge with, an approach more appropriate than that described on the web site associated with [13].

Sieving the entire tree to the same level is known to be sub-optimal in terms of segmentation performance but has the advantage of maintaining idempotency. Segmentation can also be achieved by constructing a scale-tree from the sieve decomposition and then pruning the tree appropriately [4]. Therefore, examining the scale trees of figure 7 provides an indication of the sieves' segmentation potential. The CCS gives a balanced tree structure with a low tree depth, with similar numbers of child in each branch and few, if any, leaf nodes off the root. In contrast, the PCS tree resembles that of the VAMS, with a high tree depth and some leaves close to the root as fewer regions are removed at each



(a) GSAOCS (b) CCS

(c) VAMOCS (d) PCS

(e) random

Figure 6. Average Precision-Recall curves for 100 images from the Berkeley Dataset.

step. Another factor which influences the segmentation performance is the merging rule. For the PCS and PCOCS the colour of all extrema was changed to that of their closest neighbour. However, the use of other merging rules, such as those suggested by Salembier and Garrido [15], may provide better results, especially at higher scales.

To provide an indication of the robustness of the segmentations to image noise, the Lily image was corrupted by mixed impulsive and Gaussian noise and sieved as before, see figure 8. Comparing these results with the noise-free ones of figure 4 shows that all sieves are influenced by noise to some degree, even at high scales.

The final aspect to be considered is the computational complexity, as assessed by the processing times. A comprehensive evaluation of these was undertaken in [8] for

Method	F-measure (R,P)	Regions
Human	0.79 (0.70,0.90)	-
GSAOCS	0.46 (0.50,0.44)	300
CCS	0.49 (0.61,0.41)	30
VAMS	0.40 (0.51,0.32)	100
PCS	0.48 (0.65,0.38)	300
PCOCS	0.48 (0.62,0.39)	30
VAMOCS	0.51 (0.61,0.43)	50
Random	0.38(0.80,0.25)	300

Table 4. F-measure and number of regions for P-R curves of figure 6.

VAMS, VAMOCS, CCS and GSAOCS. The PCS implementation used here is 2-3 times slower than the CCS, due to the multiple hull calculations. One reason for this is that the local convex hull is constructed at each scale rather than updated. Storing and updating the hull has the potential of reducing the complexity of the PCS, PCOCS and the CCS.

4. Conclusions

A new colour morphology sieve has been described and evaluated. The PCS uses the geometry of the local convex hull to construct a scalar surface in which extrema can be identified. Compared to the CCS, fewer extreme regions result from this approach to extrema definition. In addition, the PCS allows the use of closings to flatten homogenous regions. Result show that although the number of extrema is more realistic, this does not translate into an improved segmentation performance and this is supported by an analysis of the scale-tree. Furthermore, although the use of closings reduced the number of regions it did not result in the same benefit as for the VAMOCS.

Notwithstanding, the PCS produced the best overall performance in terms of noise removal indicating that there is some merit in its extremum definition. Areas of further work include the development of more advanced tree pruning and rules for region merging algorithms, with the aim of further improving the segmentation performance.

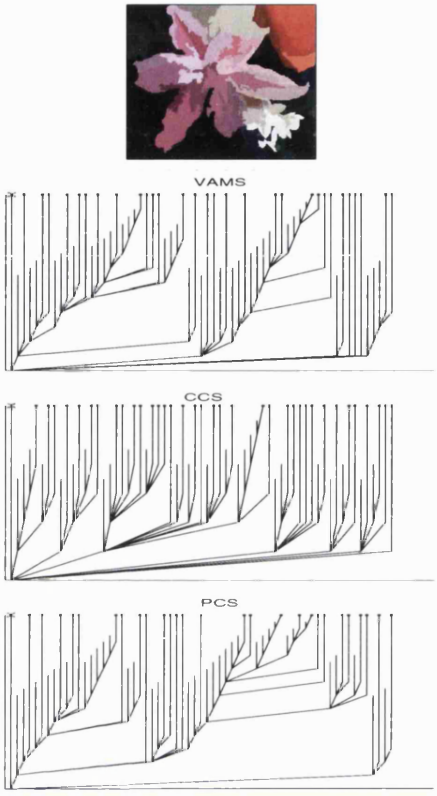


Figure 7. Colour sieve scale-trees for simplified Lily image (top).

References

[1] S.T. Acton and D.P. Mukherjee. Scale-space classification using area morphology. *IEEE Trans. Image Processing*, 9(4):623–635, April 2000.

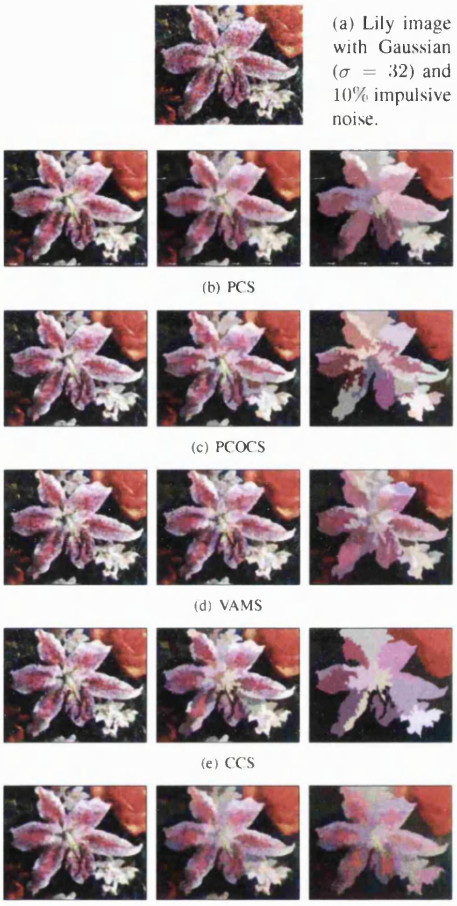
[2] J.A. Bangham, R.Harvey, P.D. Ling, and R.V. Aldridge. Morphological scale-space preserving transforms in many dimensions. *Journal of Electronic Imaging*, 5(3):283–299, July 1996.

[3] D.R. Charles. *Algorithmic and learning based filtering techniques with application to colour image noise suppression and enhancement*. Phd thesis, Machine Vision Group, Department of Physics, Royal Holloway, University of London, 2003.

[4] R.J. Chen and B.C. Chieu. Three-dimensional morphological pyramid and its application to color image sequence coding. *Signal Processing*, 44(2):163–180, 1995.

[5] A.N. Evans. Vector area morphology for motion field smoothing and interpretation. *IEE Proc. Vision, Image and Signal Processing*, 150(4):219–226, August 2003.

[6] D. Gatica-Perez, C. Gu, M.T. Sun, and S. Ruiz-Correa. Extensive partition operators, gray-level connected operators, and region merging/classification segmentation algo-



(a) Lily image with Gaussian (σ = 32) and 10% impulsive noise.

(b) PCS

(c) PCOCS

(d) VAMS

(e) CCS

(f) GSAOCS

Figure 8. Sieve results for Lily image corrupted by mixed noise. Area = 10 (left), 100 (centre) and 1000 (right).

rithms: theoretical links. *IEEE Trans. Image Processing*, 10(9):1332–1345, September 2001.

[7] S. Gibson, R. Harvey, and G.D. Finlayson. Convex colour sieves. In *Proc. 4th International Conference on Scale Space Methods in Computer Vision*, volume LNCS 2695, pages 550–563, 2003.

[8] D. Gimenez and A.N. Evans. Colour morphological scale-spaces for image segmentation. In *Proc. British Machine Vision Conference*, pages 909–918, September 2005.

[9] R. Harvey, A. Bosson, and J.A. Bangham. The robustness of some scale-spaces. In *Proc. British Machine Vision Conference*, 1997.

[10] P.T. Jackway and M. Deriche. Scale-space properties of the multiscale morphological dilation erosion. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(1):38–51, January 1996.

[11] T. Lindeberg. *Scale-space theory in computer vision*. Kluwer, 1994.

[12] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[13] D.R. Martin, C.C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, colour and texture cues. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(5):530–549, May 2004.

[14] K.R. Park and C.N. Lee. Scale-space using mathematical morphology. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(11):1121–1126, November 1996.

[15] P. Salembier and L. Garrido. Binary partition tree as an efficient representation for image processing, segmentation and information retrieval. *IEEE Trans. Image Processing*, 9(4):561–576, April 2000.

[16] P. Salembier and J. Serra. Flat zones filtering, connected operators, and filters by reconstruction. *IEEE Trans. Image Processing*, 4(8):1153–1160, August 1995.

[17] P.E. Trahanias and A.N. Venetsanopoulos. Color edge detection using vector order statistics. *IEEE Trans. Image Processing*, 2(2):259–264, Apr. 1993.

[18] N. Young and A.N. Evans. Image noise reduction using attribute morphology filters. In *Proc. 6th IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, June 2003.

[19] N. Young and A.N. Evans. Psychovisually tuned attribute operators for pre-processing digital video. *IEE Proc. Vision, Image and Signal Processing*, 150(4):277–286, October 2003.

Colour Morphological Scale-Spaces for Image Segmentation

David Gimenez and Adrian N. Evans
Department of Electronic and Electrical Engineering
University of Bath
Bath, BA2 7AY, UK
eepdg@bath.ac.uk , A.N.Evans@bath.ac.uk

Abstract

Morphological scale-spaces have become an important tool for analysing greyscale images. However, their extension to colour images has proven elusive until recently. In this paper an original evaluation of two recently proposed colour sieves is presented, both algorithmically and in terms of their computational and segmentation performance. A new colour sieve structure is also proposed, motivated by the relative advantages of the two sieves previously studied. A quantitative evaluation of the segmentation performance using a set of images with human ground truth from the Berkeley dataset shows the new method to produce the best segmentation performance.

1 Introduction

Morphological scale-space filters provide an attractive alternative to diffusion methods for the hierarchical analysis, segmentation [16] and classification [1] of greyscale images. Current greyscale morphology scale-spaces can be considered to belong to two categories: those based on the use of fixed structuring elements [13, 4] and those that employ area operators [2, 1]. Both classes are implemented by the application of successive openings and closings of increasing scale to produce a tree-based image representation. Of the two approaches the latter is the most attractive as it obeys the property of strong causality and, unlike the former, does not require any a priori knowledge of the shape of objects present.

In common with other morphological methods, major difficulties are encountered when trying to extend morphological scale-spaces to colour or other multichannel images as vector values cannot be placed in an unambiguous order. There have been some attempts to propose definitions for colour openings and closings for fixed structuring elements [3] that could be employed for scale-spaces but it is the advantages of the area morphology approach that have lead to its extension to colour receiving recent attention. In particular, in 2003 two approaches to colour morphological scale-spaces were presented, here referred to as the convex colour sieve (CCS) [8, 9] and the vector area morphology sieve (VAMS) [5, 6]. These sieves employ connected operators which operate by altering the colour of connected regions of constant colour, called flat zones, to produce regions that show some correspondence with image objects and, in the greyscale case, have been formally related to segmentation algorithms based on region merging/classification [7].

This paper presents a comparison of the two colour sieve methods, both algorithmically and in terms of evaluating their extrema definition, processing speed and segmentation performance. In addition, a new colour sieve structure is proposed that seeks to combine the relative merits of the two methods. Section 2 briefly reviews greyscale sieves and describes the CCS and VAMS algorithms. Section 3 presents the new colour sieve structure and a comprehensive performance evaluation is undertaken in section 4, including a quantitative evaluation of the sieves' segmentation performance using the Berkeley segmentation dataset. Finally, discussion and conclusions are given in section 5.

2 Colour Morphological Scale-Spaces

A greyscale area open-close (AOC) sieve can be formed by successive area openings and closings of increasing scale [1] and, for the image X, is defined by

$$AOC_i(X) = \varphi_i^a \gamma_i^a (\varphi_{i-1}^a \gamma_{i-1}^a (\ldots (\varphi_2^a \gamma_2^a (\varphi_1^a \gamma_1^a (X))))) \quad (1)$$

where γ_i^a and φ_i^a are respectively area openings and closings to an area limit i. This sieve structure is also known as an $\mathscr{M}$ sieve [2] and forms a tree as the maxima and minima are merged with their closest greyscale neighbour. In a similar manner, the closings can be performed before the openings, giving rise to an area close-open (ACO) or $\mathscr{N}$ sieve. Combining the maxima and minima in this manner ensures the property of strong causality, with no new extrema being generated as the scale increases. Alternatively, maxima and minima can be treated separately to give the max- and min-trees of [14].

Algorithmically, area morphological scale-space algorithms follow these steps:

1. Identification of extrema regions;
2. Merge all scale 1 extrema regions with their nearest neighbour;
3. Repeat step 2 with increasing scale until only 1 region remains.

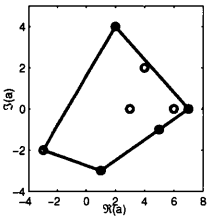
The CCS and the VAMS were developed independently and first presented within a few days of each other at Scale-Space 2003 [8] and the 2003 IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing [5] respectively. Although they differ in some aspects, the two techniques follow the same general algorithm:

1. Identification of extrema regions;
2. Merge all scale 1 extrema regions with their nearest neighbour;
3. Repeat steps 1 and 2 until no extrema are found at current scale;
4. Repeat steps 1 to 3 with increasing scale until only 1 region remains.

Comparing the greyscale and colour scale-space algorithms, it can be seen that the latter requires an additional stage (step 3) to ensure itempotence. This is because the process of merging in vector spaces can result in the creation of new extrema in the area of influence of the merged regions. Both the CCS and the VAMS use the Euclidean distance to select the closest region for merging with in Step 2, although they differ in how ties are resolved: the CCS uses luminance, then individual colour channels while the VAMS uses scan order. They also both only process extrema as it is not possible to differentiate between maxima and minima for vector values. Therefore, their main difference lies in the mechanism for determining extreme regions in Step 1 of the algorithm and to provide a comparison between the two techniques their approaches are discussed in detail below.

0+0i	4+2i	4+2i	5−1i	−3−2i
0+0i	2+4i	5−1i	5−1i	6+0i
2+4i	2+4i	5−1i	7+0i	6+0i
0+0i	2+4i	1−3i	3+0i	0+0i
0+0i	0+0i	1−3i	0+0i	−2+0i

(a)



(b)

18.00	18.00	18.00	24.50	**29.00**
18.00	31.50	24.50	24.50	14.00
31.50	31.50	24.50	31.00	14.00
12.67	31.50	**32.50**	31.00	15.50
12.67	12.67	**32.50**	15.50	9.00

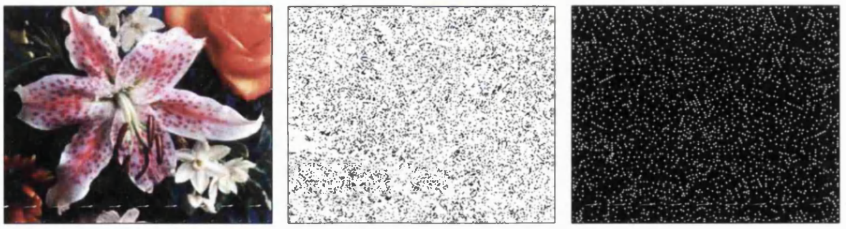
(c)

Figure 1: Colour sieves example using 8nn connectivity. (a) Complex image, (b) convex hull for $5 - 1i$ region and (c) VAMS scalar image with extrema marked in bold.

2.1 Determination of Extrema

Unlike greyscale sieves, the multivariate data associated with colour sieves cannot be unambiguously ordered and as the CCS and VAMS use different extrema definitions they produce different extrema for a given image. As colour sieves generate additional extrema at each scale as a result of the region merging process the behaviour of a colour sieve is in a large part determined by the proportion of image regions marked as extreme, which in turn depends on the extrema definition. If a high proportion of extrema regions is found then the repetition of steps 1 and 2 of the algorithm will result in few, if any, regions of area less than the current scale surviving while a low proportion of extrema regions will leave a significant proportion of the image untouched until relatively large scales. In addition, the proportion of extrema regions has a significant effect on the processing time.

The approach to extrema definition adopted by the CCS is one based on a local convex hull. For a set S of points in d-dimensional space, the convex hull is the smallest convex polygon containing all the points of S. The CCS forms a local convex hull for each pixel and its connected neighbours and then defines the pixel as extreme if it lies on the edge of the hull. This approach has the advantage that the topology of the local hull is unaffected by linear axes transformations and monotonic scaling but can result in large proportion of extrema. For example, consider the complex image shown in figure 1(a). The local convex hull for the $5 - 1i$ region in figure 1(b) shows the region to be extreme as it lies on the edge of the hull. However, examination of the local convex hull for the other regions in the image shows that they are all extreme. Part of this problem results from degenerate cases: a d-dimensional hull requires at least $(d + 1)$ different points to be non-degenerate and although an approach to reduce the number of extrema for degenerate cases was presented in [8], they can still correspond to a significant proportion of the total extrema. This result is confirmed by figure 2 which shows the initial extrema for the 186×230 colour test image Lily. The CCS result in figure 2(b) classifies the majority of the image as extrema, a finding confirmed by other images. With 4nn connectivity the number of dimensions is reduced and the CCS, if anything, finds even more extrema. Therefore the action of the CCS is very aggressive and the total number of regions will rapidly decrease with increasing area size. However, classifying the majority of pixels as extreme is counter-intuitive as extrema are associated with outlying values. In addition, there are many connected extrema which cannot simply be explained as alternating maxima and minima, which cannot be differentiated in colour sieves.



(a) (b) (c)

Figure 2: CCS and VAMS extrema (white pixels) using 8nn connectivity. (a) Original image Lily, (b) CCS extrema and (c) VAMS extrema.

The VAMS uses reduced ordering to form a scalar image in which colour extrema correspond to regional maxima. In the scalar image, the value of each pixel is initially given as the sum of the vector differences between the pixel and its connected neighbours. The scalar values for all the pixels in each flat zone are then summed and normalised by the region's area and this value is assigned to all pixels within the region, see figure 1(c). Finally, extrema are identified as the maxima of the scalar image. In figure 1(c) there are only two extreme regions corresponding to the flat zones with values $-3-2i$ and $1-3i$ in figure 1(a) and although it is theoretically possible for neighbouring regions to be marked as extreme if they have exactly the same scalar value in practice this rarely happens. The extrema found by the VAMS for the Lily image are presented in figure 2(c) and confirm the findings of figure 1 that the VAMS produces far fewer extrema than the CCS. As the VAMS extrema correspond to pixels whose colour is very different from their neighbours (either isolated regions or at positions of high gradient), they can therefore be considered as "true" extrema. However, if extrema are viewed as seeds from which the image is altered through merging, then a low extrema proportion may result in much of the image being unaffected by the sieving process until larger scales.

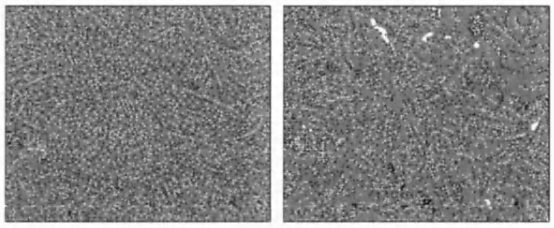
3 Vector area morphology open-close sieve

Summarising the previous section, the CCS has many extrema and an aggressive sieving action while the VAMS finds fewer, more meaningful extrema and is less aggressive. In addition, while the convex hull used by the CCS is quite inflexible with regards to its extrema definition, changes can be made to the VAMS structure to try and combine the advantages of the two methods. To this end, the vector area morphology open-close sieve (VAMOCS) is proposed that applies both area openings and closings to the scalar surface produced by the VAMS. While the maxima in the scalar surface correspond to image extrema, local minima mark regions that are closer in value to their neighbouring regions than other regions in their connected neighbourhood. Processing the minima essentially merges regions that are in relatively flat parts of the image in a manner reminiscent to that of [15], which introduces a bound Δ on the allowable greyscale fluctuations within a flat zone. Providing the merging is handled sensibly closings can increase the numbers of seeds without adversely affecting the segmentation performance. Here, the merging rules

Figure 3: Colour sieve results for Lily test image using 8nn connectivity. Top to bottom: CCS, VAMOCS (openings only), VAMOCS (closings only) and VAMOCS (combined openings and closings). Left to right: area = 10,100,1000 and 10000.

proposed by Salembier and Garrido [14] are used for closings. The VAMOCS algorithm also has an additional minor modification in which the sum of scalars for the pixels within a flat zone is normalised by its perimeter rather than its area. As the distance between all pixels within a flat region is zero, this approach effectively calculates the average vector difference per unit perimeter for all pixels on the perimeter of a region and makes the sieving action less dependent on the complexity of the regions' geometry.

Figure 3 shows the results of sieving the Lily image at selected scales using the CCS and the VAMOCS. Also shown are the VAMOCS results using openings and closings only (rows 2 and 3 of figure 3 respectively). Separate openings and closings both preserve colour edges while creating large flat regions although their filtering actions differ, with the former removing outliers and the latter leaving islands of extrema as the relatively flat regions are extended. The combined VAMOCS results shows the openings and closings to compliment each other, producing a colour sieve with an action similar in aggressiveness to the CCS. However, comparing the VAMOCS extrema in figure 4(a) with those of figure 2 shows the VAMOCS to produce many fewer extrema than the CCS and roughly equal in number to the greyscale extrema shown figure 4(b). The VAMOCS therefore appears to have achieved its aims and a comprehensive performance analysis of all the colour sieves is undertaken in the next section.



(a) (b)

Figure 4: Extrema for Lily test image produced by (a) VAMOCS and (b) greyscale AOC sieve. Maxima shown in white and minima in black.

4 Performance Evaluation

To evaluate the performance of the colour sieves, a number of criteria were used including processing time and the proportion of the image regions defined as extreme. One of the most useful applications of these sieves is colour image segmentation and an initial study of the segmentation performance of the CCS was undertaken in [9]. Here, the Berkeley segmentation dataset is used to provide the basis for a quantitative evaluation. The colour sieves were implemented in C++ using an approach based on the pixel-queue algorithm [12] and then run as mex files under Matlab. Eight nearest neighbours connectivity and the Euclidean distance metric were used for all sieves.

The first aspect to be considered is the processing time, given in figure 5, which also includes the greyscale AOC (GS-AOC) sieve for comparative purposes. As the processing times are related to the number of extreme regions, each of which has to be merged and updated, the percentage of extrema for each sieve is also plotted, with exact values for selected scales given in table 1. As expected, the greyscale sieve has the lowest processing time, although the VAMS is only fractionally slower for areas < 100. The CCS has the highest processing time reflecting its high proportion of extrema, although its rapid reduction in the total number of regions produces a relatively constant processing time for areas > 100. In contrast, the processing time for the VAMS increases with scale and at

scale	1	2	10	50	100	500	1000
VAMS	2209	1054	151	21	8	1	1
	41207	37199	27547	19660	16102	8024	6871
CCS	33616	12852	2211	456	223	44	23
	41207	16402	2878	572	279	53	25
GS-AOC	4062	1973	424	110	63	11	7
	41207	36553	28457	20697	17040	9114	6150
VAMOCS	4388	2353	476	112	16	8	9
	41207	33882	14326	3856	1845	298	132

Table 1: Variation of proportion of image extrema with scale. Fractions give the number of extreme regions the total number of regions.

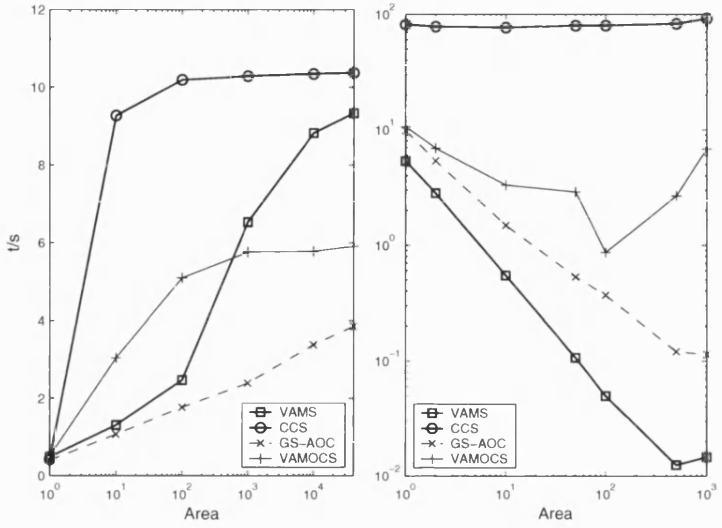


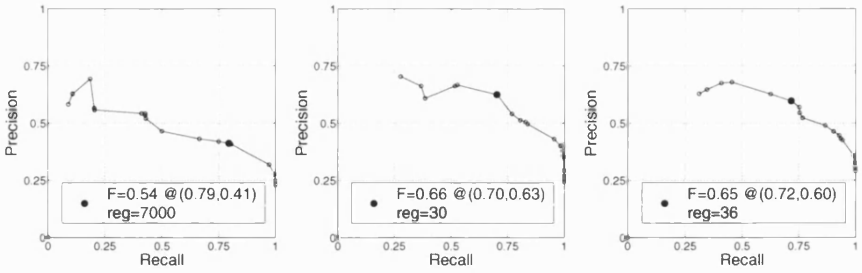
Figure 5: Processing times and proportion of extrema regions for Lily test image.

high scales approaches that of the CCS. The VAMOCS is more expensive than the VAMS for scales < 100 as it has more extrema to process. However, as it reduces the total number of regions more rapidly (see table 1), larger scales require little extra processing.

An objective measure of segmentation performance is obtained by using the Berkeley Segmentation Dataset and Benchmark available at http://www.cs.berkeley.edu/projects-/vision/grouping/segbench [10]. The ground truth for each image in the dataset is given as the collection of boundaries produced by all human subjects. A quantitative performance measure is provided by precision-recall (P-R) curves, where precision is the probability of boundary pixels being correctly identified and recall is the amount of boundary pixels detected. The F-measure is given by the harmonic mean of precision and recall along the curve, with its maximum providing a single measure segmentation performance [11]. In addition, it allows both edge- and region-based methods to be compared and avoids the trivial cases where the GCE of [10] gives zero error. Unlike [9] where the evaluation takes the most representative set of regions near the root of the tree, trimming branches to the level (attribute) required, here the entire tree is sieved to the same level. Although suboptimal in terms of segmentation performance, this allows a focus on the global threshold while maintaining itempotency.

P-R curves for 100 images from the dataset were generated by sieving each image with increasing scale until the total number of regions was $\leq n$ and plotting precision and recall for a range of values of n. A fixed number of regions was used in preference to a constant area (or other attribute) since it is less dependent on image content and also allows comparison with other attributes. This approach is also compatible with the dataset definition where a small $(2-30)$ number of equally important regions is suggested.

The P-R curves for one image from the dataset produced by the GS-AOC sieve, CCS and VAMOCS are shown in figure 6, with the point producing the maximum F-measure marked in bold. To help visualise these results, figure 7 presents the corresponding human



(a) (b) (c)

Figure 6: P-R curves for (a) GS-AOC Sieve, (b) CCS and (c) VAMOCS. The maximum F-measure and number of regions at which it occurred are also shown.

and colour sieve segmentations. These results are for just one image and a more comprehensive comparison is given by averaging the P-R curves and peak F-measure for all 100 images, see table 2. The target number of regions at which the peak F-measure occurred is also shown as a technique that can achieve a high F-measure with the minimum number of regions is preferable to one with more regions. The results in table 2 show that both the CCS and VAMOCS out-perform the greyscale sieve, although the VAMS does not. The VAMOCS has the highest average F-measure of 0.51, achieved with a target number of 50 regions. Table 2 also presents a set of results for the contrast attribute. These follow the same trend as the area attribute with the VAMOCS result being 0.02 higher than the CCS, albeit with an increased number of regions.

Method	Attribute	F-measure (R,P)	Regions
GS-AOC	Area	0.46 (0.50,0.44)	300
CCS		0.49 (0.61,0.41)	30
VAMS		0.40 (0.51,0.32)	100
VAMOCS		0.51 (0.61,0.43)	50
GS-AOC	Contrast	0.47 (0.55,0.41)	1000
CCS		0.50 (0.65,0.41)	30
VAMS		0.35 (0.42,0.30)	500
VAMOCS		0.52 (0.65,0.43)	90

Table 2: Figure of merit comparison using the Berkeley dataset

5 Discussion and conclusions

Two recently proposed colour morphological scale-space sieves have been evaluated algorithmically and in terms of their definition of extrema. The link between proportion of image regions defined as extreme and the processing speed was also investigated. The CCS was found to produce large numbers of extrema and show an aggressive sieving action whereas the VAMS has fewer extrema and therefore a lower computational cost. To

(a) (b) (c)

Figure 7: Colour sieve segmentation of Koala image. (a) Original image and collection of human segmentations, (b) and (c) CCS and VAMOCS results for different target number of regions (top) and best result (bottom).

combine their relative advantages, the VAMOCS was proposed. The VAMOCS works by also processing image minima which correspond to regions in "nearly flat" parts of the image. For scales >100, the VAMOCS has the lowest processing time of all colour sieves despite its aggressive action.

A major application area of greyscale connected sieves has been image segmentation and to assess any advantage gained by applying colour sieves to this task a quantitative evaluation of its segmentation performance was undertaken using the methodology of [11]. The new VAMOCS produced the best average segmentation performance over 100 images, showing the benefits conferred by using colour. Although its overall performance falls short of that of state-of-the-art colour segmentation techniques such as the combined brightness/colour/texture gradients of [11], the potential of the VAMOCS for segmentation has been demonstrated. An improved segmentation performance can be achieved by more advanced post-processing of the colour tree and algorithms for this, and the development of improved attributes, are areas of current research.

References

[1] S.T. Acton and D.P. Mukherjee. Scale-space classification using area morphology. *IEEE Trans. Image Processing*, 9(4):623–635, April 2000.

[2] J.A. Bangham, R. Harvey, P.D. Ling, and R.V. Aldridge. Morphological scale-space preserving transforms in many dimensions. *Journal of Electronic Imaging*, 5(3):283–299, July 1996.

[3] M.L. Comer and E.J. Delp. Morphological operations for color image processing. *Journal of Electronic Imaging*, 8(3):279–289, 1999.

[4] E.R. Dougherty, J.T. Newell, and J.B. Pelz. Morphological texture-based maximum likelihood pixel classification based on local granulometric moments. *Pattern Recognition*, 25(10):1181–1198, 1992.

[5] A.N. Evans. Extending area morphology to multivariate images. In *Proc. 6th IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, June 2003.

[6] A.N. Evans. Vector area morphology for motion field smoothing and interpretation. *IEE Proc. Vision, Image and Signal Processing*, 150(4):219–226, August 2003.

[7] D. Gatica-Perez, C. Gu, M.T. Sun, and S. S.Ruiz-Correa. Extensive partition operators, gray-level connected operators, and region merging/classification segmentation algorithms: theoretical links. *IEEE Trans. Image Processing*, 10(9):1332–1345, September 2001.

[8] S. Gibson, R. Harvey, and G.D. Finlayson. Convex colour sieves. In *Proc. 4th International Conference on Scale Space Methods in Computer Vision*, volume LNCS 2695, pages 550–563, 2003.

[9] S. Gibson, R. Harvey, and G.D. Finlayson. Evaluating a colour scale-space. In *Proc. British Machine Vision Conference*, June 2003.

[10] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[11] D.R Martin, C.C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, colour and texture cues. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(5):530–549, May 2004.

[12] A. Meijster and M.H.F. Wilkinson. A comparison of algorithms for connected set openings and closings. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(4):484–494, April 2002.

[13] K.R. Park and C.N. Lee. Scale-space using mathematical morphology. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(11):1121–1126, November 1996.

[14] P. Salembier and L. Garrido. Binary partition tree as an efficient representation for image processing, segmentation and information retrieval. *IEEE Trans. Image Processing*, 9(4):561–576, April 2000.

[15] P. Salembier, A. Oliveras, and L. Garrido. Antiextensive connected operators for image and sequence processing. *IEEE Trans. Image Processing*, 7(4):555–570, April 1998.

[16] P. Salembier and J. Serra. Flat zones filtering, connected operators, and filters by reconstruction. *IEEE Trans. Image Processing*, 4(8):1153–1160, August 1995.

ARTICLE IN PRESS

Available online at www.sciencedirect.com
ScienceDirect
Computer Vision and Image Understanding xxx (2007) xxx–xxx
Computer Vision and Image Understanding
www.elsevier.com/locate/cviu

An evaluation of area morphology scale-spaces for colour images

David Gimenez, Adrian N. Evans *

Department of Electronic and Electrical Engineering, University of Bath, BA2 7AY, UK

Received 1 September 2006; accepted 27 February 2007

Abstract

Area morphological scale-spaces are widely used for hierarchical image analysis and segmentation. Despite their advantages, their extension to colour images has been restricted by the lack of an explicit order relationship for vector values. This paper presents a theoretical evaluation of two recently proposed colour sieves and their properties. It is also demonstrated that the extrema definition used by a colour sieve determines both the aggressiveness of its sieving action and its processing speed. A new colour sieve structure is introduced that attempts to capture the relative advantages of the two sieves previously studied. An objective study of the noise reduction performance of these colour sieves is presented. The segmentation performance is also analysed using the methodology provided by the Berkeley Segmentation Dataset and Benchmark, both in terms of the overall segmentation performance and its robustness to image noise. The new colour sieve is shown to have the best overall segmentation performance, and to be the most robust.

Keywords: Colour scale-spaces; Morphological sieves; Image segmentation

1. Introduction

Openings and closings are two of the fundamental operations in mathematical morphology and, as such, are the building blocks for many other morphological operations [1]. The performance of the opening and closing operations depends in a large part on the structuring element used. If some a priori knowledge of the size, shape and orientation of objects within the image is available, then this can be used to select the structuring elements for the filtering operation. An alternative approach that has application when little or no a priori shape information is available is to use area morphology, which removes light or dark structures of a given number of pixels regardless of their shape or orientation. The concept of area operators for greyscale images has been considered by a number of authors, see for example [2,3,4]. However, as pointed out in [5], area openings and closings were first described as "a new type of opening operators (NOP) and closing operators (NCP)" by Cheng and Venetsanopoulos [6]. Although the application of area operations to greyscale images was initially limited by their high computational cost, the development of more efficient algorithms by Vincent [3], Salembier and Sera [2], and more recently Meijster and Wilkinson [5], has removed this barrier for many applications. A more generalised approach to area morphology, in which other attributes can be used to control the filtering action, was developed by Breen and Jones [7]. In their approach, both increasing and nonincreasing criteria can be accommodated using attribute openings and thinnings, respectively.

The application of a succession of area openings and closings of increasing scale results in scale-space filters that do not suffer from some of the problems associated with those derived using fixed structuring elements, for example those of [8] and [9], or diffusion methods. In particular, area morphological scale-spaces have the property of strong causality which ensures that not only are no new edges created as scale increases but also that the positions of the existing edges do not drift through scale [4,10]. Scale-spaces based on connected operators act on regions of constant intensity, termed flat zones. At small scales, these flat zones can be thought of as image noise [11] while

* Corresponding author.
E-mail addresses: eepdg@bath.ac.uk (D. Gimenez), A.N.Evans @ bath.ac.uk (A.N. Evans).

1077-3142/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2007.02.004

Please cite this article in press as: D. Gimenez, A.N. Evans, An evaluation of area morphology scale-spaces for colour images, Comput. Vis. Image Understand. (2007), doi:10.1016/j.cviu.2007.02.004

at larger scales they show some correspondence with image objects. Consequently, one of the main applications of connected set morphology is image segmentation [12]. Indeed, Gatica-Perez et al. have established a formal relationship between connected operators and segmentation algorithms based on region merging [13].

The extension of scale-spaces based on both structural and connected openings and closings to colour images is not straightforward. This is because their underlying operations rely on pixel ordering and, for vector-valued images, no unambiguous ordering exists. Various definitions of structural erosions and dilations for colour images have been proposed using, for example, reduced ordering [14], conditional ordering [15], a combination of the two [16] or other approaches such as graph decimation [17]. Any of these can be used to generate colour scale-spaces. However, as they employ fixed structuring elements they suffer from the same disadvantages as their greyscale counterparts. Therefore, recent research interest has been directed at the development of colour scale-spaces based on area morphology. Two such approaches to colour sieves are the convex colour sieve (CCS) of Gibson et al. [18,19] and the vector area morphology sieve (VAMS) of Evans [20,21]. These sieves were developed independently and work by processing regions of constant colour (the flat zones) using connected operators. Unlike the connected filters of [22], they do not introduce any new colours during the processing.

In this paper, the two approaches to colour scale-spaces of the CCS and the VAMS are analysed both algorithmically and experimentally. It is shown that the extrema definition used by each colour sieve is of critical importance for its subsequent performance. The vector area morphology open–close sieve (VAMOCS) represents an attempt to combine the advantages of the CCS and the VAMS. The VAMOCS was first proposed in [23], where its processing speed and segmentation performance was considered. This paper presents an extended discussion of the colour sieves' algorithm and properties, an evaluation of their colour noise reduction performance and a study of the robustness of their segmentation results.

The remainder of this paper is organised as follows. A review of area morphology scale-spaces and a description of the colour sieves and their extrema definitions, algorithm and properties is given in Section 2. Section 3 presents an experimental evaluation of the colour sieves. Included here is an analysis of the sieves' noise reduction capability, an objective evaluation of their segmentation performance and its robustness to image noise. Conclusions are drawn in Section 4.

2. Area morphological colour sieves

2.1. Introduction

Area openings γ_λ^a and closings φ_λ^a for a greyscale image I can be defined by

$$\gamma_\lambda^a(I) = \bigvee_{B \in A_\lambda} (I \circ B) \tag{1}$$

and

$$\varphi_\lambda^a(I) = \bigwedge_{B \in A_\lambda} (I \bullet B) \tag{2}$$

respectively, where A_λ is the set of connected subsets with area $\geqslant \lambda$. Although subsequent definitions make use of the threshold decomposition [10,24]. Eqs. (1) and (2) explicitly show that area operators select the most appropriately shaped structuring element of a given area at each pixel position. An area opening (resp. closing) removes all the light (resp. dark) structures from an image that have an area of less than λ.

Alternating area openings and closings of increasing scale can be combined in an alternating sequential filter (ASF) structure to give the greyscale area open–close (AOC) and area close-open (ACO) scale-spaces [10]. For a scale size of λ, the AOC and ACO scale-spaces are defined by

$$AOC_\lambda(I) = \varphi_\lambda^a\gamma_\lambda^a(\varphi_{\lambda-1}^a\gamma_{\lambda-1}^a(\ldots(\varphi_2^a\gamma_2^a(\varphi_1^a\gamma_1^a(I))))) \tag{3}$$

and

$$ACO_\lambda(I) = \gamma_\lambda^a\varphi_\lambda^a(\gamma_{\lambda-1}^a\varphi_{\lambda-1}^a(\ldots(\gamma_2^a\varphi_2^a(\gamma_1^a\varphi_1^a(I))))). \tag{4}$$

These scale-spaces are also known as M- and N-sieves [4] and produce a tree structure as each regional maximum and minimum is merged with its closest greyscale neighbour. Choosing to combine the openings and closings in this order produces a sieve structure that is guaranteed not to produce any new extrema as the scale increases. Although the sieves described by Eqs. (3) and (4) are ASF, not all ASF have the properties of sieves [4]. It should also be noted that, since opening and closing do not commute, the AOC and ACO sieves are not guaranteed to produce identical results [10]. An alternative approach after Salembier and Garrido is to process the maxima and minima separately using max-trees and min-trees [25].

Algorithmically, the greyscale area morphological scale-space sieves described by Eqs. (3) and (4) can be considered to have the following main steps:

(1) Identify all regional extrema;
(2) Merge all scale 1 regional extrema with their nearest neighbour;
(3) Repeat step 2 with increasing scale, up to scale = λ.

The two connected colour sieves, the CCS and the VAMS, were both first presented in 2003, see [18] and [20], respectively. Although developed independently, both sieves have the same steps in their algorithms:

(1) Identify all regional extrema;
(2) Merge all scale 1 regional extrema with their nearest neighbour;
(3) Repeat steps 1 and 2 until no extrema exist at current scale;

Please cite this article in press as: D. Gimenez, A.N. Evans, An evaluation of area morphology scale-spaces for colour images, Comput. Vis. Image Understand. (2007), doi:10.1016/j.cviu.2007.02.004

(4) Repeat steps 1–3 with increasing scale, up to scale = λ.

Unlike greyscale sieves, neither the CCS nor the VAMS distinguish between maxima and minima, both of which are classified as colour extrema. Therefore in step 1 both colour sieves simply identify the vector extrema for subsequent processing. This contrasts with the greyscale case where there are two types of extrema (maxima and minima) to process. As the colour sieves identify and process all the extrema at each scale, regardless of whether they can be classified as maxima or minima, their action is comparable to that of the greyscale ASF-based sieves.

The merging process in step 2 is analogous to that of the greyscale sieve and sets the colour of each extremum to the colour of the closest neighbouring region, measured using a suitable distance metric. The main difference between the greyscale and colour sieve algorithms is the additional step required to ensure that any new extrema created by the colour merging process are themselves merged if their area is less than the current scale. The inclusion of step 3 thus ensures that idempotence is maintained. However, as step 3 requires the previous 2 steps to be repeated at each scale, the performance of the colour sieves in a large part depends on their extrema definition and this is examined in detail below.

2.2. Colour extrema

Treating each pixel of a colour image as a vector produces an image representation in which the pixels cannot easily be ordered. This situation is unlike that of greyscale images and has obvious implications for colour sieves. As colour sieves not only determine extrema in the initial step of their algorithms, but also produce new extrema as a consequence of the merging process in step 2, the proportion of regions defined as extreme is of critical importance. For example, a high proportion of image extrema is characteristic of a sieve with an aggressive sieving as few regions smaller than the current area will survive at each scale. Alternatively, if extrema are viewed as seed positions for image simplification, then a definition that produces few extrema will not significantly alter the image until larger scales. The proportion of extrema also directly relates to the processing time.

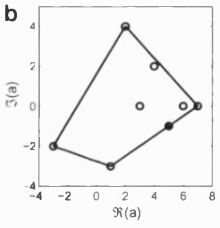
The CCS determines if a region is extreme by using its local convex hull [18]. Here, a pixel is extreme if it lies on the edge of the convex hull for the pixel and its four- or eight-connected neighbours. Figs. 1(a) and (b) show an example two-dimensional image (expressed using complex numbers) and the local convex hull for the $5 - 1i$ region, respectively. This approach has the advantage that the hull topology is invariant to monotonic scaling and linear axes transformations. However, it typically results in a large proportion of regions being defined as extreme, which does not conform with the intuitive interpretation of extrema as outliers. For example, the CCS will identify all the regions in the complex image of Fig. 1(a) as extrema, a situation that cannot be simply explained by the inability of colour sieves to distinguish between maxima and minima.

A different approach to vector extrema is used by the VAMS [20,21]. The first stage of the VAMS is to replace the vector at each pixel position by the sum of the distances to its connected neighbours, calculated using a L_p norm. Next, for each flat region containing more than one pixel the mean value with respect to the norm is calculated and assigned to all its pixels. Finally, the maxima in the scalar surface define the vector extrema; these are essentially those regions whose sum of distances to its connected

a

0+0i	4+2i	4+2i	5−1i	−3−2i
0+0i	2+4i	5−1i	5−1i	6+0i
2+4i	2+4i	5−1i	7+0i	6+0i
0+0i	2+4i	1−3i	3+0i	0+0i
0+0i	0+0i	1−3i	0+0i	−2+0i

b



c

18.00	18.00	18.00	24.50	**29.00**
18.00	31.50	24.50	24.50	14.00
31.50	31.50	24.50	31.00	14.00
12.67	31.50	**32.50**	31.00	15.50
12.67	12.67	**32.50**	15.50	9.00

d

6.00	4.50	4.50	5.16	**9.67**
6.00	**6.64**	5.16	5.16	3.50
6.64	**6.64**	5.16	3.88	3.50
5.43	**6.64**	5.91	3.88	3.88
5.43	5.43	5.91	3.88	3.00

Fig. 1. Colour sieves example using 8nn connectivity. (a) Two-dimensional vector image expressed using complex numbers, (b) convex hull for $5 - 1i$ region, (c) and (d) VAMS scalar image normalised by area and perimeter, respectively. Extrema regions marked in bold in (c) and (d).

Please cite this article in press as: D. Gimenez, A.N. Evans, An evaluation of area morphology scale-spaces for colour images, Comput. Vis. Image Understand. (2007), doi:10.1016/j.cviu.2007.02.004

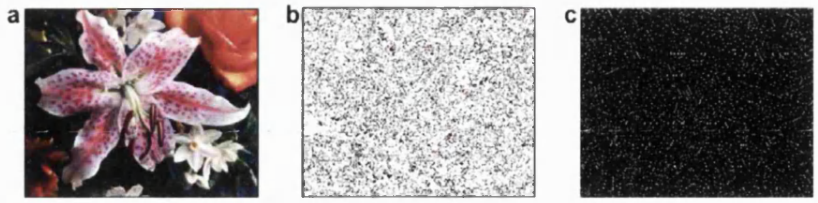


Fig. 2. Colour extrema for the Lily test image. (a) Original image, (b) CCS extrema and (c) VAMS extrema. Extrema are shown in white and are for 8 nn connectivity.

neighbours is greater than that of any neighbouring region. The approach produces significantly fewer extrema than the CCS, as illustrated by the scalar image for the example complex image shown in Fig. 1(c), which only has two extreme regions.

As an additional comparison Fig. 2 shows the initial extrema produced by the CCS and the VAMS for the Lily test image. Fig. 2(b) shows that the CCS classifies over 80% of the flat regions in the image as extreme whereas for the VAMS the proportion drops to less than 6%. For comparison, approximately 10% of the regions in the corresponding luminance image are extreme. It should be reiterated that, unlike greyscale sieves, colour sieves create and merge additional extrema as scale increases, so the extrema definition not only determines the initial extrema but also the numbers of new extrema at each scale.

Whilst the local convex hull employed by the CCS essentially makes a binary extreme/non-extreme decision for each region, the scalar surface used by the VAMS allows more flexibility to be introduced to the sieve structure. The vector area morphology open–close sieve (VAMOCS) is a new sieve structure that aims to produce an improved performance by also processing the minima in the VAMS' scalar surface. The minima regions are flat zones that have a smaller colour difference to their connected neighbours than any of their neighbouring regions. Applying closings to the scalar surface has the effect of merging regions whose colour is similar to the surrounding colours. This approach is similar in spirit to that of [26], where the use of a bound on the greyscale fluctuations was introduced in response to the observation that visual entities may not be strictly flat. However, care must be taken when merging regions using closings and here the merged region is assigned the colour vector of the largest flat zone [25], provided it is also the least extreme.

The main motivation of the VAMOCS is to provide an improved segmentation performance by increasing the number of extrema, that act as seeds for the process of image simplification through the merging of flat zones. A further refinement that improves the segmentation performance is to use the perimeter of the flat region when calculating the mean L_p norm, rather than the area [23]. The result of this normalisation for the example complex image is shown in Fig. 1(d) and the overall effect is to reduce the influence of the regions' geometry.

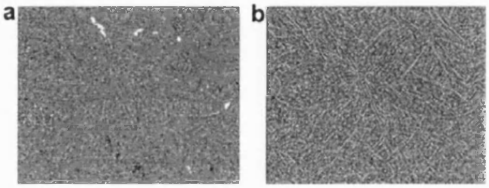


Fig. 3. Maxima (black) and minima (white) for Lily test image. (a) Greyscale extrema and (b) VAMOCS extrema.

Fig. 3 shows the greyscale maxima and minima for the Lily image and those produced by the perimeter-normalised variant of the VAMOCS. Both techniques define approximately 10% of the regions as extreme. Fig. 4 shows the Lily image sieved to scales 10, 100, 1000 and 1000 by the VAMS and the VAMOCS. Also shown are the results of closing the minima in the scalar surface. As expected, the closing operation creates large flat zones while leaving the maxima regions. This action complements that of the VAMS and when combined in the VAMOCS results in a colour sieve whose sieving action is similar in aggression to the CCS but with a fraction of the extrema.

2.3. Algorithm and properties

The colour sieve algorithm given in Fig. 5 uses an approach based on the pixel queue algorithm of Vincent [3,27]. In the first step (line 1), all extrema regions are identified and placed in the appropriate list. These lists are a series of first-in-first-out (FIFO) queues, where each queue contains all the extrema of a given area. The queues are then processed in increasing order until the desired area λ is reached. The algorithm contains two subtleties, both resulting from the colour merging process which can create and destroy extrema in the local neighbourhood of the merged regions. New extrema are detected and added to the appropriate list (line 7). However, as the area of the new extrema may be less than the current sieve area, it is necessary to ensure that no extrema exist up to the current sieve scale in line 3. Merely checking the list for the current scale would allow extrema with areas of less than the current scale to survive thus contradicting the property of idempotency. Similarly, the merging process can also cause

Please cite this article in press as: D. Gimenez, A.N. Evans, An evaluation of area morphology scale-spaces for colour images, Comput. Vis. Image Understand. (2007), doi:10.1016/j.cviu.2007.02.004

(a) Area = 10.

(b) Area = 100.

(c) Area = 1000.

(d) Area = 10000.

Fig. 4. Colour sieve results for Lily test image using 8nn connectivity. VAMS (left column), VAMOCS (closings only) and VAMOCS (combined openings and closings).

1. Extract all extremum flat zones and place in list *extrema(area)*, according to their area
2. For *area* = 1 to λ:
3. While *extrema(i)* $\forall$ *i* = 1, 2, ..., *area* are not empty
4. For each flat zone in *extrema(area)*
5. If flat zone is still an extremum
6. Merge region with its closest neighbour.
7. Append any new extrema created by merging process to appropriate list.
8. Else remove flat zone from *extrema(area)*

Fig. 5. Algorithm used for CCS, VAMS and VAMOCS.

existing extrema to become non-extreme. Rather than detecting these cases and then searching for and removing them from the appropriate list, a simple alternative is to check that a region is still extreme (line 5) before the merging is performed.

The implementations of the colour sieve algorithm for the CCS and VAMS only differ in how the extrema are defined. For the VAMOCS the situation is slightly more complex as the test in line 5 must determine whether the extremum corresponds to a maximum or a minimum in the scalar surface and then perform the appropriate merging.

Greyscale sieves have certain properties, not all of which directly translate to their colour analogues. In the colour sieve algorithm, idempotence is guaranteed by the inclusion of the while loop in line 3 that ensures that no extrema of area up to, and including, the current size survive. In greyscale morphology, openings and closings preserve the order relationship between images. For colour images, the absence of an unambiguous vector ordering means that increasingness can not be demonstrated and, further, is not a meaningful criterion. However, like greyscale sieves, as scale increases the colour sieves reduce the total number of flat regions, do not introduce any new colours and have the property of strong causality.

It is also noted that the colour sieves are not invariant to the order of processing the extrema. This is because colour extrema are analogous to greyscale maxima and minima and when two or more adjacent extrema occur the order of processing affects the merging result. The equivalent case in greyscale is that of the M- and N-sieves of Bangham et al. [4] (also known as the ACO and AOC scale-spaces [10]) in which the results depend on whether the minima are processed before the maxima or visa versa. In the colour sieve algorithm given in Fig. 5 all the existing extrema at the current scale are processed first, followed by the newly created extrema in order of increasing scale.

3. Experimental evaluation

The CCS, VAMS and VAMOCS were implemented using the algorithm detailed in Fig. 5. The connectivity was eight nearest neighbours and the Euclidean distance, applied in the RGB colour space, was used for determining the closest neighbour for merging step. Although some results for other colour spaces and distance metrics are reported, we focus on RGB as techniques that perform well in this colour space can easily be extended to other multichannel images. Fig. 6 shows the variation in the proportion of extrema and processing time with scale for each colour sieve. For comparison the greyscale AOC sieve was applied to the luminance component of the colour image, although is should be noted that more efficient algorithms than the pixel queue exist for the greyscale case [5]. It can be seen that the proportion of regions that are classified as extreme is as would be expected from the analysis in the previous section. Those sieves with high proportion of extrema also more rapidly reduce the total number of image regions as the extrema are merged. The CCS has both the highest proportion of extrema and processing times, although the latter is almost constant for larger scales as very few regions remain. For scales below 1000 the VAMS is faster than the VAMOCS as it has fewer

Please cite this article in press as: D. Gimenez, A.N. Evans, An evaluation of area morphology scale-spaces for colour images, Comput. Vis. Image Understand. (2007), doi:10.1016/j.cviu.2007.02.004

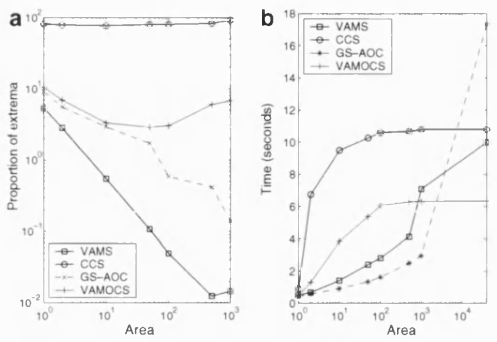


Fig. 6. Colour sieve performance for Lily image. (a) Proportion of regions classified as extreme and (b) processing times.

extrema to merge. However, for areas >1000 the more rapid reduction in the total number of regions by the VAMOCS results in this situation being reversed. The results in Fig. 6 are for the area-normalised VAMS and perimeter-normalised VAMOCS, although in practice the alternative normalisations produce very similar results. It should also be noted that the choice of colour space has more influence on the proportion of extrema and processing times of the VAMS and VAMOCS than for the CCS.

The remainder of this evaluation has two main foci. Firstly, the ability of the colour sieves to reduce image noise and, secondly, the segmentation performance that can be achieved on noise-free and corrupted images. The emphasis on segmentation is appropriate as this is one of the main application areas of greyscale connected sieves [25]. To provide a quantitative evaluation of segmentation performance the ground truth segmentations and evaluation methodology provided by the Berkeley Segmentation Dataset and Benchmark[1] is used [28,29].

3.1. Noise reduction

At smaller scales, the extrema regions removed by the colour sieves correspond to image noise. Although the noise reduction performance of the colour sieves may not be comparable with that produced by state-of-the-art filters, scale-space sieves have the advantage of removing noise without introducing any positional shifts to the remaining edges. To investigate the noise reducing capabilities of the colour sieves, the Lily test image was corrupted by Gaussian and impulsive noise. Three noise levels were used for each noise type and the noise was additionally classified as independent or correlated, with a correlation coefficient $\rho = 0.5$ [14,30]. The corrupted images were sieved through a range of scales and, for each scale, the normalised mean square error (NMSE) recorded. The noise reduction capability of each sieve is then given by the minimum NMSE, see Table 1. In this table, the results for the VAMS and VAMOCS are further classified according to whether area or perimeter was used to normalise the scalar surface before finding the extrema regions.

The results in Table 1 show that, apart from the lowest level of Gaussian noise, all colour sieves reduce image noise to some degree. For all levels of impulsive noise the perimeter-normalised VAMS is the best performing sieve and the CCS the worst. For both the area- and perimeter-normalised variants, the VAMS produces better results than the corresponding VAMOCS, for both correlated and independent impulsive noise. For Gaussian noise the situation is not so clear. At the lowest level of Gaussian noise ($\sigma^2 = 100$) all the colour sieves introduce more distortion than the original noise. With a medium level of independent Gaussian noise ($\sigma^2 = 10^3$) the CCS produces the lowest NMSE. For all other Gaussian noise levels the area-normalised VAMS and VAMOCS are the best and next-best performing sieves, respectively, and the perimeter-normalised VAMS and VAMOCS are the worst two. The scale at which the minimum NMSE occurred increased with noise level for both Gaussian and impulsive noise, with the former requiring significantly larger scales for comparable distortion. Whether the noise was correlated or independent appears to have little influence on scale.

A further measure of the noise reduction performance of the colour sieves is provided by the mean chromaticity error (MCRE). As the MCRE is unaffected by any errors in brightness, the scale that produces the minimum MCRE will not necessarily correspond with the scale of the minimum NMSE. The approach adopted here was to use the MCRE to give a measure of the chromatic error at the scale that produced the minimum NMSE. These results are given in Table 2. The MCRE results for impulsive noise follow a similar trend to the NMSE results, with the best and worst performing sieves the same as before. However, for 1% of impulsive noise the MCRE for all sieves was higher than for the unfiltered image. For Gaussian noise the results are less clear, although either the CCS or the area-normalised VAMS has the lowest MCRE for all but one noise case.

The results in Tables 1 and 2 show the colour sieves to be more robust to impulsive noise than Gaussian and this is confirmed by Fig. 7 which shows the best performing sieve results for medium noise levels of correlated impulsive and Gaussian noise. Fig. 7 shows a discernable improvement in image quality for both noise types, accompanied by no distracting colour bleeding at edges. Using the L_1 norm as the distance metric reduced the errors for impulsive noise and increased them for Gaussian noise. Finally, a similar relative ranking was obtained when the sieves were applied in the L*a*b* colour space, although the absolute average errors were slightly lower.

[1] Available at http://www.cs.berkeley.edu/projects/vision/grouping/segbench.

Please cite this article in press as: D. Gimenez, A.N. Evans, An evaluation of area morphology scale-spaces for colour images, Comput. Vis. Image Understand. (2007), doi:10.1016/j.cviu.2007.02.004

Table 1
NMSE ($\times 10^{-2}$) for colour sieves applied to Lily image

Noise model	Sieve type					
	None	CCS	VAMS		VAMOCS	
			Area	Perimeter	Area	Perimeter
1% Independent impulsive	0.411	0.403 (2)	0.307 (2)	0.241 (2)	0.319 (2)	0.268 (2)
1% Correlated impulsive	0.798	0.406 (2)	0.318 (2)	0.251 (2)	0.330 (2)	0.279 (2)
10% Independent impulsive	3.849	0.847 (3)	0.667 (3)	0.600 (3)	0.695 (3)	0.639 (3)
10% Correlated impulsive	7.665	0.948 (3)	0.762 (3)	0.714 (3)	0.784 (3)	0.750 (3)
40% Independent impulsive	15.401	2.135 (5)	1.908 (5)	1.874 (6)	1.959 (5)	1.959 (6)
40% Correlated impulsive	30.560	2.867 (9)	2.397 (9)	2.321 (9)	2.478 (9)	2.495 (9)
Independent Gaussian ($\sigma^2 = 10^2$)	0.527	0.780 (2)	0.732 (2)	0.677 (2)	0.743 (2)	0.700 (2)
Correlated Gaussian ($\sigma^2 = 10^2$)	0.528	0.768 (2)	0.713 (2)	0.659 (2)	0.724 (2)	0.688 (2)
Independent Gaussian ($\sigma^2 = 10^3$)	4.716	3.163 (11)	3.236 (14)	3.381 (10)	3.217 (9)	3.418 (8)
Correlated Gaussian ($\sigma^2 = 10^3$)	4.689	3.206 (10)	3.158 (10)	3.285 (10)	3.178 (9)	3.291 (9)
Independent Gaussian ($\sigma^2 = 10^4$)	31.054	10.514 (200)	9.430 (380)	11.701 (70)	10.281 (60)	12.706 (60)
Correlated Gaussian ($\sigma^2 = 10^4$)	30.887	10.708 (100)	9.672 (130)	11.443 (110)	10.165 (200)	11.337(130)

Figures in brackets give the scale at which the minimum NMSE occurred.

Table 2
MCRE ($\times 10^{-2}$) for colour sieves applied to Lily image

Noise model	Sieve type					
	None	CCS	VAMS		VAMOCS	
			Area	Perimeter	Area	Perimeter
1% Independent impulsive	0.371	1.236	0.587	0.382	0.706	0.626
1% Correlated impulsive	0.437	1.237	0.590	0.382	0.709	0.625
10% Independent impulsive	3.520	1.844	1.148	0.954	1.347	1.301
10% Correlated impulsive	4.228	1.885	1.162	0.955	1.354	1.311
40% Independent impulsive	13.898	2.948	2.742	2.767	2.924	3.207
40% Correlated impulsive	16.757	3.280	2.778	2.699	3.104	3.315
Independent Gaussian ($\sigma^2 = 10^2$)	6.293	5.531	5.827	5.947	5.830	5.927
Correlated Gaussian ($\sigma^2 = 10^2$)	4.830	4.536	4.683	4.716	4.702	4.748
Independent Gaussian ($\sigma^2 = 10^3$)	17.562	11.254	12.338	13.6281	12.536	13.680
Correlated Gaussian ($\sigma^2 = 10^3$)	14.195	9.905	11.019	11.791	10.913	11.524
Independent Gaussian ($\sigma^2 = 10^4$)	34.961	19.274	16.842	23.949	20.167	23.240
Correlated Gaussian ($\sigma^2 = 10^4$)	28.687	15.705	18.733	21.410	15.976	17.975

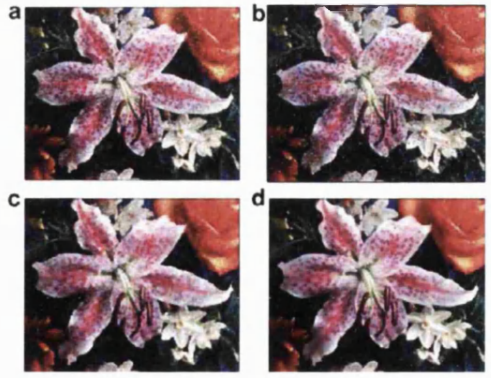


Fig. 7. Noise results for the Lily test image. (a) and (b) Lily image corrupted by correlated Gaussian noise ($\sigma^2 = 1000$) and 10% impulsive noise, respectively. (c) Area-normalised VAMS result for (a) and (d) Perimeter-normalised VAMS result for (b).

3.2. Colour image segmentation

To provide an objective measure of segmentation performance of the colour sieves the Berkeley Segmentation Dataset and Benchmark is used. The dataset contains a large number of natural images, each of which has been hand-segmented by human observers [28], while the benchmark provides a methodology for quantifying segmentation performance [29]. This methodology uses the human segmentations to provide ground truth boundaries and characterises the segmentation performance by precision–recall (P–R) curves. Here, precision measures the probability that a detected boundary pixel is contained in the ground truth and recall is the probability of detecting a true boundary pixel [29]. To generate the P–R curves for the colour sieves each test image was sieved until the total number of regions was less than a predetermined threshold number and the precision and recall plotted for a range of threshold values. A fixed number of regions was used in preference to the scale parameter as this is less dependent

Please cite this article in press as: D. Gimenez, A.N. Evans, An evaluation of area morphology scale-spaces for colour images, Comput. Vis. Image Understand. (2007), doi:10.1016/j.cviu.2007.02.004

on image content and also allows comparison with other attributes. The approach is also compatible with the dataset ground truth images which contain a small number of equally important regions.

Fig. 8 gives example P–R curves produced by the CCS and the perimeter-normalised VAMOCS for the colour test image shown in Fig. 8(a) using the collection of human segmentations in Fig. 8(b). The Berkeley benchmark also uses the F-measure to characterise the segmentation performance of an algorithm with a single number. The F-measure is the harmonic mean of precision and recall and the maximum F-measure position on the P–R curve measures the best segmentation result. Boundary maps corresponding to the maximum F-measure positions for the CCS and the VAMOCS are given in Fig. 8(e) and (f), respectively.

The procedure described above was repeated for 100 images from the dataset and the average F-measures for each colour sieve found, see Table 3. Results for both the area and contrast attributes were recorded, as were the threshold number of regions at which the maximum F-measure occurred. The latter is important as when two segmentation techniques have the same F-measure, the result with the fewest regions is preferable. In Table 3, only the results for the area- and perimeter-normalised variants of the VAMS and VAMOCS, respectively, are given as these were higher then the alternative normalisations.

Table 3
Average F-measure for 100 images from the Berkeley dataset achieved by the CCS, area-normalised VAMS, perimeter-normalised VAMOCS and a greyscale AOC ASF applied to the luminance component

Method	Attribute	F-measure	No. Regions
CCS	Area	0.49	30
VAMS		0.40	60
VAMOCS		0.51	70
GS-AOC		0.37	3000
CCS	Contrast	0.50	30
VAMS		0.35	500
VAMOCS		0.52	90
GS-AOC		0.38	5000

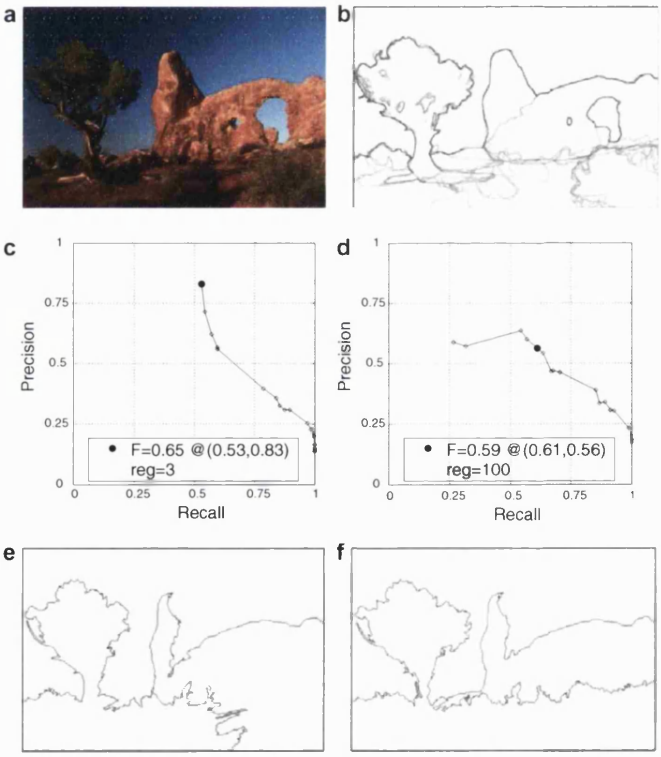


Fig. 8. Example segmentation evaluation. (a) Colour test image, (b) collection of human segmentations. (c) and (d) Precision–recall curves produced by the CCS and VAMOCS, respectively. (e) and (f) Boundary maps that produced the maximum F-measure in (c) and (d).

Please cite this article in press as: D. Gimenez, A.N. Evans, An evaluation of area morphology scale-spaces for colour images, Comput. Vis. Image Understand. (2007), doi:10.1016/j.cviu.2007.02.004

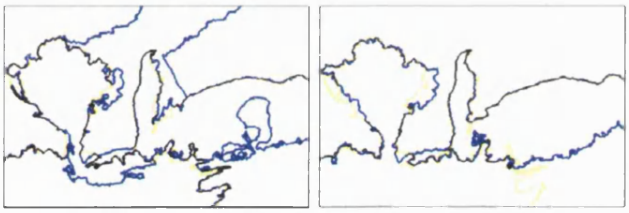
The results for the area attribute show that the CCS has a better segmentation performance than the VAMS and this performance is achieved with fewer regions. However, when closings are also considered the highest average F-measure of 0.51 for the VAMOCS results. The F-measures for the contrast attribute show the same pattern. Here, the VAMOCS result of 0.52 is again 0.02 above that of the CCS. The benefit of incorporating colour within the sieve structure is clearly demonstrated by the fact that the average F-measure for the colour sieves is 0.09 above the average greyscale sieve result, and the best performing colour sieve is 0.14 better. Somewhat counterintuitively, when the sieves were applied in L*a*b* colour space the resulting segmentation performance was slightly worse.

In practical applications images are rarely noise-free and therefore the robustness of any segmentation algorithm to image noise is an important performance criterion. This was investigated by applying the colour sieves to noisy images and generating P–R curves by using the best noise-free segmentation result as the ground truth. The F-measure then provides a measure of how closely the segmentation results of the noisy images match the best noise-free segmentation results. As an illustration, Fig. 9 shows the best noisy segmentation results for the example image of Fig. 8(a). This example is very difficult to evaluate sub-

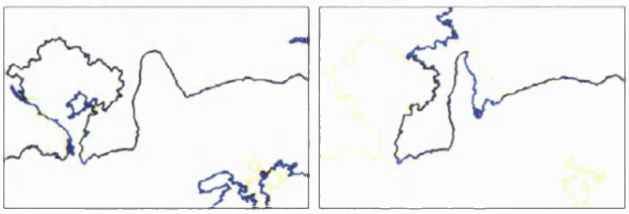
Table 4
Robustness of colour sieve segmentations to (independent) image noise

Noise	CCS	VAMS	VAMOCS
0.1% Impulsive	0.670	0.478	0.610
1% Impulsive	0.590	0.452	0.577
10% Impulsive	0.543	0.456	0.568
Gaussian ($\sigma^2 = 100$)	0.537	0.436	0.558
Gaussian ($\sigma^2 = 1000$)	0.490	0.452	0.499
Gaussian ($\sigma^2 = 10000$)	0.431	0.415	0.436

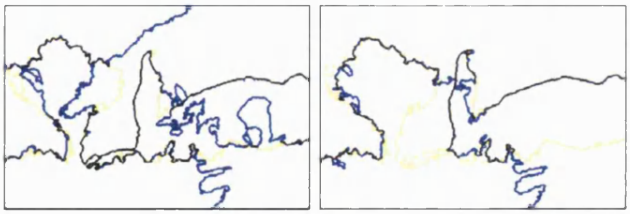
Average F-measures between noisy and noise-free segmentations for 100 images from dataset.



(a) CCS results.



(b) VAMS results.



(c) VAMOCS results.

Fig. 9. Segmentation results for Fig. 8(a) corrupted by 10% independent impulsive noise (left column) and independent Gaussian noise ($\sigma^2 = 1000$) (right column), shown in blue. Ideal result shown in yellow and overlap shown in black. (For interpretation of the references to colors in this figure legend, the reader is referred to the web version of this paper.)

Please cite this article in press as: D. Gimenez, A.N. Evans, An evaluation of area morphology scale-spaces for colour images, Comput. Vis. Image Understand. (2007), doi:10.1016/j.cviu.2007.02.004

jectively, reiterating the need for a quantitative evaluation methodology. Table 4 gives the average F-measures for 100 images from the dataset corrupted by various levels of independent impulsive and Gaussian noise. These results show that the VAMS is the least robust for all noise types and levels. The CCS has the greatest robustness for 0.1 and 1% impulsive noise while for all other cases the VAMOCS is the most robust.

4. Conclusions

Greyscale scale-space sieves have proved a useful tool for image analysis, and in particular image segmentation. To extend connected sieves to colour images new techniques must be developed as, for example, simply applying a greyscale sieve to individual channels can produce erroneous results. The CCS and VAMS are two recently proposed connected sieves for colour image that treat each pixel as a vector value. This paper has presented an analysis of these two sieves, showing that they have a common algorithm but differ in their definition of extrema regions. This is significant as it is demonstrated that the proportion of regions defined as extreme has a major influence on the performance of the sieves: the CCS has a high proportion of extrema and an aggressive sieving action whilst the VAMS has a low proportion of extrema that limits its segmentation performance.

The VAMOCS is a new colour sieve structure that also applies closings to the scalar surface of the VAMS. This increases the proportion of regions that are classified as extreme to approximately 10% of the total. Experimental results show the processing time of the VAMOCS is still comparable to that of the VAMS despite its more aggressive sieving action. The use of both area and perimeter to normalise the scalar surface used by the VAMS and VAMOCS was also investigated. An evaluation of the noise reducing abilities has shown the VAMS to be the most effective of the colour sieve structure. The best results were achieved when the scalar surface was normalised by perimeter for impulsive noise and area for Gaussian noise.

Connected sieves are widely used for image segmentation and a quantitative evaluation of the segmentation performance of the colour sieves was undertaken using the Berkeley Segmentation Dataset and Benchmark. Results show the VAMOCS to produce the best overall segmentation performance results. Analysis of the robustness of the colour sieves' segmentation results has also been undertaken, showing that the VAMOCS is also the most robust. It should be noted that the segmentation performance achieved by all colour sieves is not competitive with state-of-the-art colour segmentation techniques. However, if the sieve is viewed as a pre-processing step that transforms the image into a tree structure containing many possible segmentations, then the potential of these colour sieves can be seen. In particular, the VAMOCS performs significantly better than a greyscale sieve applied to the luminance component. How to select sets of regions that give improved segmentations is an area of current research. Although the focus of this paper has been on colour, the sieves' algorithms are generic in nature and can also be applied to other multichannel data, such multi- and hyper-spectral images.

References

[1] J. Serra, Image Analysis and Mathematical Morphology, Academic Press, 1982.
[2] P. Salembier, J. Serra, Flat zones filtering, connected operators, and filters by reconstruction, IEEE Transactions on Image Processing 4 (8) (1995) 1153–1160.
[3] L. Vincent, Morphological area openings and closings for grey-scale images, in: Shape in Picture: Mathematical Description of Shape in Grey-level Images, 1993, pp. 196–208.
[4] J.A. Bangham, R. Harvey, P. Ling, R. Aldridge, Morphological scale-space preserving transforms in many dimensions, Journal of Electronic Imaging 5 (3) (1996) 283–299.
[5] A. Meijster, M.H.F. Wilkinson, A comparison of algorithms for connected set openings and closings, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (4) (2002) 484–494.
[6] F. Cheng, A.N. Venetsanopoulos, An adaptive morphological filter for image processing, IEEE Transactions on Image Processing 1 (1992) 533–539.
[7] E.J. Breen, R. Jones, Attribute openings, thinnings, and granulometries, Computer Vision and Image Understanding 64 (1996) 377–389.
[8] E. Dougherty, J. Newell, J. Pelz, Morphological texture-based maximum likelihood pixel classification based on local granulometric moments, Pattern Recognition 25 (10) (1992) 1181–1198.
[9] K.R. Park, C.N. Lee, Scale-space using mathematical morphology, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (11) (1996) 1121–1126.
[10] S.T. Acton, D.P. Mukherjee, Scale-space classification using area morphology, IEEE Transactions on Image Processing 9 (4) (2000) 623–635.
[11] N. Young, A.N. Evans, Psychovisually tuned attribute operators for pre-processing digital video, IEE Proceedings-Vision, Image and Signal Processing 150 (5) (2003) 277–286.
[12] J. Crespo, R.W. Schafer, J. Serra, C. Gratin, F. Meyer, The flat zone approach: a general low-level region merging segmentation method, Signal Processing 62 (1) (1997) 37–60.
[13] D. Gatica-Perez, C. Gu, M.T. Sun, S. Ruiz-Correa, Extensive partition operators, gray-level connected operators, and region merging/classification segmentation algorithms: theoretical links, IEEE Transactions on Image Processing 10 (9) (2001) 1332–1345.
[14] M.L. Comer, E.J. Delp, Morphological operations for color image processing, Journal of Electronic Imaging 8 (3) (1999) 279–289.
[15] V. Caselles, G. Sapiro, D.H. Chung, Vector median filters, inf-sup operations, and coupled pde's: Theoretical connections, Journal of Mathematical Imaging and Vision 12 (2) (2000) 109–119.
[16] L.J. Sartor, A.R. Weeks, Morphological operations on color images, Journal of Electronic Imaging 10 (2) (2001) 548–559.
[17] O. Lezoray, C. Meurie, A. Elmoataz, A graph approach to color mathematical morphology, in: Proc. 5th IEEE International Symposium on Signal Processing and Information Technology, 2005, pp. 856–861.
[18] S. Gibson, R. Harvey, and G.D. Finlayson, Convex colour sieves, in: Proc. 4th International Conference on Scale Space Methods in Computer Vision, Vol. LNCS 2695, 2003, pp. 550–563.
[19] S.E. Gibson, J.A. Bangham, and R. Harvey, Evaluating a colour scale-space, in: Proc. British Machine Vision Conference, 2003.
[20] A.N. Evans, Extending area morphology to multivariate images, in: Proc. 6th IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing, 2003.

Please cite this article in press as: D. Gimenez, A.N. Evans, An evaluation of area morphology scale-spaces for colour images, Comput. Vis. Image Understand. (2007), doi:10.1016/j.cviu.2007.02.004

[21] A.N. Evans, Vector area morphology for motion field smoothing and interpretation, IEE Proceedings-Vision, Image and Signal Processing 150 (4) (2003) 219–226.
[22] K.R. Weber, S.T. Acton, On connected filters in color image processing, Journal of Electronic Imaging 13 (3) (2004) 619–629.
[23] D. Gimenez, A.N. Evans, Colour morphological scalespaces for image segmentation, in: Proc. British Machine Vision Conference, 2005, pp. 909–918.
[24] P. Maragos, R.D. Ziff, Threshold superposition in morphological image analysis systems, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (5) (1990) 498–504.
[25] P. Salembier, L. Garrido, Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval, IEEE Transactions on Image Processing 9 (4) (2000) 561–576.
[26] P. Salembier, A. Oliveras, L. Garrido, Antiextensive connected operators for image and sequence processing, IEEE Transactions on Image Processing 7 (4) (1998) 555–570.
[27] L. Vincent, Morphological area openings and closings, their efficient implementation and applications, in: Proceedings of the "EURASIP Workshop on Mathematical Morphological and its Application to Signal Processing", May 1993, Barcelona, Spain, 1993, pp. 22–27.
[28] D.R. Martin, C.C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in: Proc. 8th Int'l Conf. Computer Vision, Vol. 2, 2001, pp. 416–423.
[29] D.R. Martin, C.C. Fowlkes, J. Malik, Learning to detect natural image boundaries using local brightness, colour and texture cues, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (5) (2004) 530–549.
[30] T. Viero, K. Oistamo, Y. Neuvo, Three-dimensional median-related filters for colour image sequence filtering, IEEE Transactions on Circuits and Systems for Video Technology 4 (2) (1994) 129–142.

Please cite this article in press as: D. Gimenez, A.N. Evans, An evaluation of area morphology scale-spaces for colour images, Comput. Vis. Image Understand. (2007), doi:10.1016/j.cviu.2007.02.004

Appendix B

Median filtering results

B.1 Scale results

Table B.1a: Median NMSE on uncorrupted image against scale

Filter	Scale	lily	autumn	boats	lenna	baboon	Sample1	average
Input		0	0	0	0	0	0	0
Mean	3*3	.8993077	.6327552	.7749131	.1780544	3.254472	.4145586	1.025677
	5*5	2.252550	1.300593	1.487389	.3870988	4.736408	.6565370	1.803429
	7*7	3.419232	1.726682	2.017243	.5888079	5.416763	.8431253	2.335309
	9*9	4.424183	2.044848	2.454838	.7782293	5.869529	1.002974	2.762433
	11*11	5.342196	2.290101	2.847332	.9581218	6.220287	1.137088	3.132521
VMF	3*3	.5736517	.5522088	.5593615	.1441110	3.545039	.3581954	.9554279
L_1	5*5	1.568008	1.298919	1.339449	.2964810	5.223104	.5595815	1.714257
	7*7	2.571204	1.752355	1.818753	.4410035	5.828420	.6689994	2.180123
	9*9	3.513388	2.037556	2.167553	.5726790	6.209375	.7487969	2.541558
	11*11	4.462639	2.230497	2.457902	.6941895	6.502597	.8116676	2.890289

Table B.1b: Median NMSE with uncorrelated Gaussian noise against scale, $\sigma^2 = 1000$

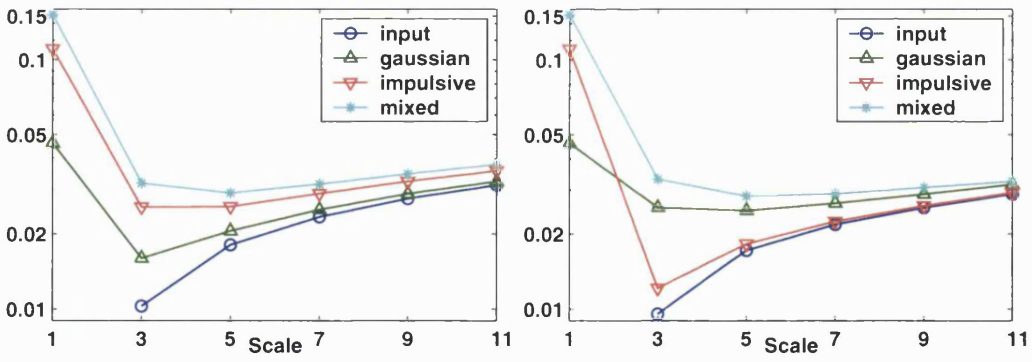
Filter	Scale	lily	autumn	boats	lenna	baboon	Sample1	average
Input		4.701120	4.644400	5.592920	4.670831	6.522292	1.594991	4.621092
Mean	3*3	1.498895	1.196555	1.440018	.7212858	4.045481	.6796941	1.596988
	5*5	2.535678	1.535355	1.763416	.6002107	5.056604	.8112590	2.050420
	7*7	3.626551	1.869329	2.195650	.7120643	5.610604	.9666942	2.496815
	9*9	4.598965	2.151849	2.595400	.8666905	6.012251	1.118208	2.890561
	11*11	5.494400	2.381013	2.969537	1.030488	6.340730	1.252909	3.244846
VMF	3*3	2.373519	2.204476	2.414520	1.633321	5.728751	.9358227	2.548402
L_1	5*5	2.595347	2.068118	2.242896	1.045057	6.065331	.8627458	2.479916
	7*7	3.319561	2.154754	2.392220	.9409591	6.229132	.8807155	2.652890
	9*9	4.109190	2.293378	2.610992	.9617278	6.401301	.9223559	2.883157
	11*11	4.915207	2.407365	2.844836	1.031213	6.577155	.9636561	3.152831

Table B.1c: Median NMSE with uncorrelated impulsive noise against scale, $\rho = 10\%$

Filter	Scale	lily	autumn	boats	lenna	baboon	Sample1	average
Input		11.57144	12.54906	11.31568	9.906269	14.88744	6.002710	11.03877
Mean	3*3	2.597932	2.406757	2.252502	1.465954	5.278619	1.309871	2.551939
	5*5	3.188370	2.203478	2.176266	.9930576	5.682638	1.150443	2.565709
	7*7	4.145130	2.387107	2.498965	1.013038	6.073280	1.234497	2.892003
	9*9	5.058864	2.611338	2.857735	1.134078	6.410938	1.358052	3.238501
	11*11	5.934084	2.818437	3.210810	1.283046	6.705591	1.480627	3.572099
VMF	3*3	.9434301	.7237716	.7781142	.2365537	4.110820	.4657314	1.209737
L_1	5*5	1.787458	1.359956	1.428925	.3414494	5.386689	.6178140	1.820382
	7*7	2.728004	1.762306	1.868574	.4760313	5.904143	.7092482	2.241385
	9*9	3.629785	2.026459	2.241538	.6014706	6.234886	.7853276	2.586578
	11*11	4.531483	2.203832	2.623292	.7240319	6.509740	.8461486	2.920823

Table B.1d: Median NMSE with uncorrelated mixed noise against scale, $\sigma^2 = 1000$ and $\rho = 10\%$

Filter	Scale	lily	autumn	boats	lenna	baboon	Sample1	average
Input		15.81624	16.69609	16.31965	14.11537	20.77618	7.431423	15.19249
Mean	3*3	3.260763	3.062971	2.927421	2.019597	6.112503	1.735410	3.186444
	5*5	3.572328	2.573601	2.507451	1.252543	6.100043	1.484833	2.915133
	7*7	4.452244	2.675696	2.739523	1.190756	6.378908	1.540766	3.162982
	9*9	5.329102	2.867700	3.062002	1.278795	6.670377	1.655044	3.477170
	11*11	6.177797	3.059564	3.396448	1.411624	6.942934	1.775721	3.794015
VMF	3*3	3.209156	2.930551	3.288621	2.248766	6.964273	1.209557	3.308487
L_1	5*5	3.073047	2.391394	2.627235	1.316880	6.579558	1.018957	2.834512
	7*7	2.661245	3.689621	2.361966	1.121176	6.536269	.9978593	2.894689
	9*9	4.366609	2.441065	2.854897	1.099787	6.616069	1.028623	3.067842
	11*11	5.142354	2.531797	3.110116	1.145400	6.743180	1.063469	3.249122



(a) Mean (b) Median

Figure B.1: Average NMSE for uncorrelated noise against mask width

Table B.2a: Median MCRE on uncorrupted image against scale

Filter	Scale	lily	autumn	boats	lenna	baboon	Sample1	average
Input		0	0	0	0	0	0	0
Mean	3*3	2.085148	1.937705	1.852019	1.624976	7.432581	2.315913	2.874724
	5*5	3.152297	2.310318	2.323390	2.016542	7.975509	3.033511	3.468595
	7*7	3.844515	2.499039	2.643944	2.292648	8.215863	3.310735	3.801124
	9*9	4.343019	2.662146	2.908003	2.520059	8.436294	3.590275	4.076633
	11*11	4.742692	2.802260	3.107753	2.714508	8.616424	3.864906	4.308091
VMF	3*3	1.539239	1.793455	1.484598	1.405205	7.197147	1.585506	2.500858
L_1	5*5	2.539734	2.282574	2.027110	1.801360	8.022183	2.332868	3.167638
	7*7	3.178197	2.455796	2.253333	2.032108	8.242383	2.427840	3.431610
	9*9	3.646443	2.589140	2.423298	2.206626	8.427275	2.535001	3.637964
	11*11	4.352296	2.811313	2.666890	2.476291	8.672151	2.776391	4.086249

Table B.2b: Median MCRE with uncorrelated Gaussian noise against scale, $\sigma^2 = 1000$

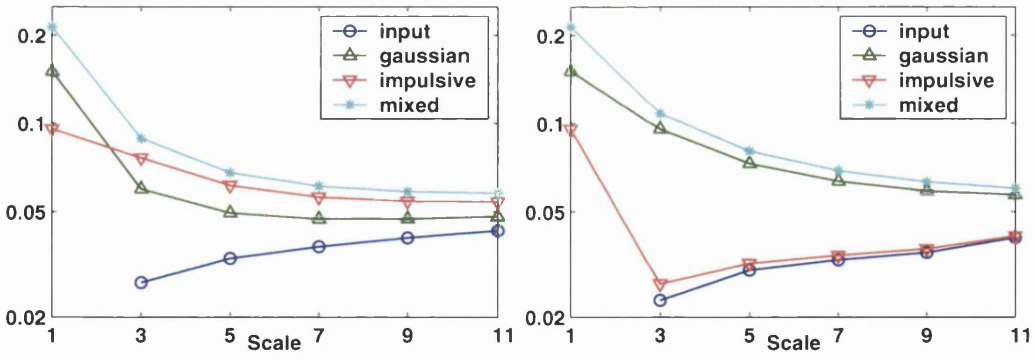
Filter	Scale	lily	autumn	boats	lenna	baboon	Sample1	average
Input		17.76449	20.04216	16.09797	12.11596	17.78781	6.533786	15.05703
Mean	3*3	6.028170	6.387754	5.662087	4.368904	9.338237	4.051665	5.972803
	5*5	4.880105	4.382537	4.323840	3.315501	8.767859	4.041750	4.951932
	7*7	4.812517	3.698667	3.950385	3.074985	8.702430	4.090883	4.721645
	9*9	4.973975	3.442115	3.856084	3.062815	8.789737	4.250413	4.729190
	11*11	5.202181	3.353333	3.853041	3.128130	8.904351	4.452662	4.815616
VMF	3*3	10.86849	11.64156	9.901854	6.904892	13.11206	4.897560	9.554404
L_1	5*5	8.045786	8.017978	7.209054	5.068352	10.89787	4.484445	7.287248
	7*7	7.063539	6.449092	5.976526	4.317547	10.02615	4.224067	6.342820
	9*9	6.630959	5.644776	5.333857	3.953585	9.661458	4.126186	5.891804
	11*11	6.370767	4.834874	4.762443	3.697315	9.405394	4.176459	5.715723

Table B.2c: Median MCRE with uncorrelated impulsive noise against scale, $\rho = 10\%$

Filter	Scale	lily	autumn	boats	lenna	baboon	Sample1	average
Input		9.961675	10.40569	9.653702	9.259682	10.41942	7.689333	9.564917
Mean	3*3	7.923060	8.291374	7.313965	5.837827	10.82139	5.508900	7.616086
	5*5	6.309910	5.819361	5.453885	4.519783	9.691044	4.999631	6.132269
	7*7	5.814288	4.753645	4.706275	4.141465	9.375303	4.811338	5.600386
	9*9	5.681735	4.266488	4.397230	4.049733	9.343610	4.832673	5.428578
	11*11	5.711532	4.016167	4.278806	4.061733	9.393232	4.942379	5.400642
VMF	3*3	2.009376	2.054329	1.762894	1.669921	7.740881	1.816579	2.842330
L_1	5*5	2.782383	2.415430	2.170042	1.921989	8.236787	2.474297	3.333488
	7*7	3.349493	2.543431	2.352752	2.115323	8.369469	2.579925	3.551732
	9*9	3.778106	2.674066	2.521790	2.272217	8.508572	2.705227	3.743330
	11*11	4.451725	2.875081	2.825457	2.530690	8.733856	2.943950	4.136464

Table B.2d: Median MCRE with uncorrelated mixed noise against scale, $\sigma^2 = 1000$ and $\rho = 10\%$

Filter	Scale	lily	autumn	boats	lenna	baboon	Sample1	average
Input		23.91495	26.16056	22.12074	18.72687	24.40714	12.78261	21.35214
Mean	3*3	9.311191	9.887553	8.558309	7.140056	11.93033	6.452109	8.879925
	5*5	6.915108	6.555381	6.159294	5.272161	10.25772	5.606887	6.794424
	7*7	6.220691	5.249684	5.305309	4.710998	9.804473	5.350292	6.106908
	9*9	5.998400	4.634056	4.935491	4.526703	9.703669	5.332203	5.855087
	11*11	5.979062	4.312056	4.764905	4.484806	9.712869	5.417469	5.778528
VMF	3*3	12.25774	13.38822	11.15818	7.99681	14.49520	5.517517	10.80228
L_1	5*5	8.949683	9.030662	7.999975	5.737248	11.60708	4.899410	8.037343
	7*7	7.756127	7.122246	6.585144	4.819372	10.50534	4.561074	6.891550
	9*9	7.135260	6.181825	5.818389	4.354374	10.00423	4.441155	6.322537
	11*11	6.707791	5.233356	5.155488	4.007929	9.631920	4.448209	6.017630



(a) Mean (b) Median

Figure B.2: Average MCRE for uncorrelated noise against mask radius

B.2 General NMSE results

Table B.3a: NMSE on uncorrupted input image

Method	lily	autumn	boats	lenna	baboon	Sample1	average
Input	0	0	0	0	0	0	0
Mean	2.252550	1.300593	1.487388	.3870988	4.736408	.6565370	1.803429
VMF L_3	1.710938	1.324257	1.393741	.3369592	5.902628	.7686652	1.906198
VMF L_2	1.585996	1.298106	1.350259	.2975318	5.289772	.5786197	1.733381
VMF L_1	1.568008	1.298919	1.339449	.2964810	5.223104	.5595815	**1.714257**
VMF L_2^2	2.301165	1.387550	1.576312	.3995372	5.191838	.6318195	1.914704
TNC_1-1*it*	1.964996	1.684304	1.650420	.3554924	5.798703	.6312200	2.014189
TNC_1-2*it*	2.228932	1.862059	1.812808	.4129570	6.046569	.6694778	2.172134
TNC_2-1*it*	1.807047	1.475952	1.515296	.3155194	5.354871	.6198383	1.848087
TNC_2-2*it*	2.082350	1.707381	1.713614	.3514351	5.599913	.6762198	2.021819
BVDF	2.981028	2.550162	2.183184	.3539301	9.166176	1.244855	3.079889
GVDF-n	2.619483	2.273336	1.906435	.3291074	7.269457	.8953252	2.548857
GVDF-$\frac{n^2}{2}$	2.358400	1.888517	1.818224	.3456101	6.061994	.7053141	2.196343
$DDF_{10\%}$	1.548813	1.933019	1.784778	.3245384	6.001752	.9928321	2.097622
$DDF_{25\%}$	1.446735	1.767144	1.629615	.3138052	5.600306	.9834980	1.956850
$DDF_{50\%}$	1.401045	1.684868	1.523757	.3064245	5.411951	.9809062	1.884825
$DDF_{75\%}$	1.369333	1.626198	1.427431	.3020553	5.314272	.9777274	1.836170
$DDF_{90\%}$	1.353055	1.590044	1.347064	.2989060	5.254801	.9737715	1.802940

Table B.3b: NMSE with uncorrelated Gaussian noise, $\sigma^2 = 1000$

Method	lily	autumn	boats	lenna	baboon	Sample1	average
Input	4.701120	4.644400	5.592920	4.670831	6.522292	1.594991	4.621092
Mean	2.551957	1.534304	1.767689	.6026375	5.057862	.8118136	2.054377
VMF L_3	5.396059	4.741130	5.359291	3.678210	10.52905	1.916176	5.269985
VMF L_2	2.769622	2.185210	2.399744	1.146950	6.281071	.8956400	2.613039
VMF L_1	2.604115	2.056731	2.242363	1.037168	6.070797	.8778145	**2.481498**
VMF L_2^2	3.129546	2.115802	2.400160	1.068373	6.105644	1.018953	2.639747
TNC_1-1*it*	2.932497	2.468770	2.604735	1.269761	6.714724	1.022571	2.835510
TNC_1-2*it*	3.102965	2.606703	2.673943	1.312384	6.832428	1.084009	2.935405
TNC_2-1*it*	2.813512	2.196232	2.376099	1.093895	6.248006	.9785904	2.617722
TNC_2-2*it*	2.888391	2.268057	2.395693	1.124532	6.405019	1.012135	2.682305
BVDF	9.843160	5.976206	7.073327	2.803725	13.30482	1.602944	6.767364
GVDF-n	7.184480	4.515576	5.055072	1.787756	9.883344	1.211753	4.939663
GVDF-$\frac{n^2}{2}$	5.634371	4.085229	4.481735	2.054339	9.077761	1.195198	4.421439
$DDF_{10\%}$	3.426744	3.815861	2.906079	1.458952	7.384233	1.112906	3.350796
$DDF_{25\%}$	2.911139	3.185407	2.446016	1.206344	6.679483	1.041634	2.911670
$DDF_{50\%}$	2.678238	2.896107	2.212627	1.105692	6.398168	.9496218	2.706742
$DDF_{75\%}$	2.550758	2.724719	2.147381	1.060028	6.255166	.9188421	2.609482
$DDF_{90\%}$	2.539515	2.705854	2.127698	1.039264	6.217204	.9324277	2.593661

Table B.3c: NMSE with partly correlated Gaussian noise, $\sigma^2 = 1000$

Method	lily	autumn	boats	lenna	baboon	Sample1	average
Input	4.688606	4.613822	5.549666	4.674576	6.505063	1.580278	4.602002
Mean	2.535024	1.532515	1.770425	0.600694	5.054343	.8128424	**2.050974**
VMF L_3	4.002237	3.270657	3.620691	2.269082	8.585874	1.564696	3.885539
VMF L_2	2.617946	2.053900	2.230470	1.038315	6.065773	.8553919	2.476966
VMF L_1	2.532432	1.946520	2.126985	.9610878	5.888832	.8317240	2.381263
VMF L_2^2	3.025831	1.988383	2.267604	.9810603	5.876573	.9626435	2.517016
TNC_1-1*it*	2.761187	2.353516	2.427388	1.148645	6.498163	1.032064	2.703494
TNC_1-2*it*	2.992898	2.547986	2.550904	1.199062	6.686877	1.152670	2.855066
TNC_2-1*it*	2.639792	2.043067	2.218105	.9879785	6.0181112	.9330630	2.473353
TNC_2-2*it*	2.709062	2.135998	2.286194	1.006631	6.173576	1.012637	2.5540167
continued							

Method	lily	autumn	boats	lenna	baboon	Sample1	average
BVDF	10.40533	7.742323	8.563254	3.151464	14.97953	1.817091	7.776498
GVDF-n	7.516311	5.202028	5.475356	1.793089	10.49885	1.442646	5.321379
GVDF-$\frac{n^2}{2}$	5.304606	3.909640	4.093446	1.701536	8.747010	1.130756	4.147832
$DDF_{10\%}$	3.409500	3.723313	2.928702	1.489936	7.352701	1.254357	3.359751
$DDF_{25\%}$	2.863966	3.066782	2.431586	1.192964	6.617934	1.075006	2.874706
$DDF_{50\%}$	2.607542	2.821475	2.213374	1.067608	6.304237	.9826416	2.666146
$DDF_{75\%}$	2.480887	2.688411	2.096688	1.006978	6.148933	.9377251	2.559937
$DDF_{90\%}$	2.424945	2.647272	2.044703	.9817743	6.077340	.9182605	2.515716

Table B.3d: NMSE with uncorrelated impulsive noise, $\rho = 10\%$

Method	lily	autumn	boats	lenna	baboon	Sample1	average
Input	11.57144	12.54906	11.31568	9.906269	14.88744	6.002710	11.03877
Mean	3.188370	2.203478	2.176266	.9930576	5.682638	1.150443	2.565709
VMF L_3	4.261952	2.711863	3.303382	1.029869	9.179133	1.263563	3.624960
VMF L_2	1.756828	1.343876	1.405334	.3247152	5.440253	.6211956	**1.815367**
VMF L_1	1.787458	1.359956	1.428925	.3414494	5.386689	.6178140	1.820382
VMF L_2^2	2.845665	1.713465	1.938708	.6172141	5.901878	.9609592	2.329648
TNC_1-1*it*	2.079084	1.640924	1.627424	.3713728	5.901109	.6746894	2.049101
TNC_1-2*it*	2.290773	1.749846	1.725869	.4055148	6.083836	.7148083	2.161775
TNC_2-1*it*	2.139821	1.482681	1.707059	.4165644	5.616318	.6933376	2.009297
TNC_2-2*it*	2.308225	1.592264	1.763780	.4283615	5.808119	.7283820	2.104855
BVDF	3.488348	2.561632	2.467833	.4330761	9.264518	1.112903	3.221385
GVDF-n	3.022754	2.288832	2.110980	.4039061	7.148101	.8236621	2.633039
GVDF-$\frac{n^2}{2}$	2.460425	1.737717	1.877258	.4092297	6.103288	0.7501253	2.223007
$DDF_{10\%}$	1.597001	2.069799	1.729191	.3745330	6.106506	.6395818	2.086102
$DDF_{25\%}$	1.498224	1.910983	1.579723	.3594775	5.718768	.6277154	1.949149
$DDF_{50\%}$	1.462799	1.830663	1.482788	.3498352	5.547935	.6233358	1.882893
$DDF_{75\%}$	1.440344	1.795938	1.410497	.3439706	5.455376	.6208641	1.844498
$DDF_{90\%}$	1.430120	1.782465	1.373502	.3421712	5.407966	.6184670	1.825782

Table B.3e: NMSE with partly correlated impulsive noise, $\rho = 10\%$

Method	lily	autumn	boats	lenna	baboon	Sample1	average
Input	7.665482	8.266423	7.530804	6.608201	9.957845	3.990191	7.336491
Mean	2.831367	1.823453	1.918497	.7547864	5.319446	.9374696	2.264170
VMF L_3	3.235246	1.971019	2.180600	.8625673	7.921243	1.127640	2.883053
VMF L_2	1.676750	1.325842	1.380302	.3133900	5.360749	.6072905	1.777387
VMF L_1	1.664581	1.320838	1.373489	.3135353	5.276491	.5794663	**1.754734**
VMF L_2^2	2.662756	1.602436	1.806921	.5499635	5.637363	.8247299	2.180695
TNC_1-1*it*	1.964301	1.566919	1.559635	.3437543	5.712775	.6491851	1.966095
TNC_1-2*it*	2.169966	1.669441	1.650051	.3743231	5.867135	.6824531	2.068895
TNC_2-1*it*	2.000840	1.433660	1.565881	.3791092	5.398124	.6396364	1.902875
TNC_2-2*it*	2.139957	1.541068	1.637109	.3883183	5.567757	.6814269	1.992606
BVDF	3.240682	3.566085	2.301992	.3755137	9.969299	1.156006	3.434930
GVDF-n	2.876025	2.299347	1.994989	.3544244	7.311461	.8356018	2.611975
GVDF-$\frac{n^2}{2}$	2.509221	1.881590	1.889902	.3748171	6.077534	.7398976	2.245494
$DDF_{10\%}$	1.560844	1.975282	1.744299	.3415765	6.069856	.6475053	2.056560
$DDF_{25\%}$	1.470804	1.821216	1.602867	.3290755	5.651473	.6401465	1.919264
$DDF_{50\%}$	1.420024	1.750279	1.489584	.3213573	5.460407	.6331402	1.845799
$DDF_{75\%}$	1.387929	1.697388	1.405330	.3173434	5.352438	.6215565	1.796997
$DDF_{90\%}$	1.380297	1.673663	1.351076	.3147818	5.299139	.6170420	1.772666

Table B.3f: NMSE with uncorrelated mixed noise, $\sigma^2 = 1000$ and $\rho = 10\%$

Method	lily	autumn	boats	lenna	baboon	Sample1	average
Input	15.81624	16.69609	16.31965	14.11537	20.77618	7.431423	15.19249
Mean	3.591832	2.568415	2.504541	1.244313	6.103495	1.480388	2.915497
VMF L_3	8.543216	6.637839	8.407466	5.275024	14.92891	2.413210	7.700944
VMF L_2	3.094287	2.438251	2.683833	1.350263	6.724294	.9841129	2.879174
VMF L_1	3.111709	2.400528	2.645157	1.315049	6.610804	1.022984	**2.851039**
VMF L_2^2	4.172276	3.130451	3.229810	1.759468	7.248612	1.662231	3.533808
TNC_1-1*it*	3.339888	2.783419	2.861739	1.510530	7.258657	1.141459	3.149282
TNC_1-2*it*	3.542355	2.934841	2.960899	1.532935	7.410507	1.201260	3.263800
TNC_2-1*it*	3.491222	2.746600	2.947149	1.462894	7.028280	1.308643	3.164131
TNC_2-2*it*	3.573737	2.860752	3.026547	1.472411	7.190653	1.359473	3.247262

continued

Method	lily	autumn	boats	lenna	baboon	Sample1	average
BVDF	10.47422	6.602046	8.171557	3.372320	14.59440	1.794311	7.501476
GVDF-n	7.688273	4.786781	5.595956	2.195620	10.61242	1.368188	5.374540
GVDF-$\frac{n^2}{2}$	6.242509	4.609776	5.224406	2.679602	10.30253	1.393678	5.075417
$DDF_{10\%}$	4.569190	4.791424	3.718889	1.827519	8.648651	1.336135	4.148635
$DDF_{25\%}$	3.917446	4.088861	3.070621	1.561823	7.710409	1.203344	3.592084
$DDF_{50\%}$	3.631102	3.765975	2.862066	1.435042	7.388616	1.166269	3.374845
$DDF_{75\%}$	3.552801	3.675948	2.795588	1.397394	7.216629	1.121612	3.293329
$DDF_{90\%}$	3.527352	3.624526	2.717258	1.372924	7.151156	1.115789	3.251501

Table B.3g: NMSE with partly correlated mixed noise, $\sigma^2 = 1000$ and $\rho = 10\%$

Method	lily	autumn	boats	lenna	baboon	Sample1	average
Input	12.04564	12.57142	12.71766	10.9726	16.03264	5.465558	11.63426
Mean	3.185765	2.145122	2.240688	1.00058	5.702156	1.222467	2.582797
VMF L_3	6.267992	4.391688	5.544861	3.746497	11.74137	2.084965	5.629561
VMF L_2	2.775361	2.149494	2.350305	1.121446	6.251726	.9080527	2.592731
VMF L_1	2.710724	2.059236	2.264397	1.047460	6.071677	.8954936	**2.508165**
VMF L_2^2	3.629925	2.564717	2.724358	1.355280	6.512721	1.356499	3.023917
TNC_1-1*it*	2.897794	2.403730	2.501720	1.202910	6.618684	1.047467	2.778717
TNC_1-2*it*	3.102960	2.559418	2.603169	1.239778	6.772452	1.135181	2.902160
TNC_2-1*it*	2.885873	2.161696	2.409622	1.105748	6.242092	1.024187	2.638203
TNC_2-2*it*	2.947953	2.249518	2.469096	1.121900	6.384444	1.077453	2.708394
BVDF	13.43325	11.12338	10.41519	3.297647	17.20370	1.873042	9.557701
GVDF-n	8.060075	5.489040	5.757010	1.903987	10.87343	1.496813	5.596725
GVDF-$\frac{n^2}{2}$	5.450256	3.927758	4.199956	1.835731	8.947662	1.186504	4.257978
$DDF_{10\%}$	4.008662	4.243937	3.377597	1.608193	7.961174	1.331285	3.755141
$DDF_{25\%}$	3.401514	3.540604	2.820383	1.303822	7.143108	1.171218	3.230108
$DDF_{50\%}$	3.136913	3.269081	2.574436	1.171907	6.797720	1.089356	3.006569
$DDF_{75\%}$	3.004876	3.146334	2.457752	1.112768	6.634588	1.052923	2.901540
$DDF_{90\%}$	2.949340	3.103094	2.407822	1.088228	6.563421	1.042622	2.859088

B.3 General MCRE results

Table B.4a: MCRE on uncorrupted input image

Method	lily	autumn	boats	lenna	baboon	Sample1	average
Input	0	0	0	0	0	0	0
Mean	3.152297	2.310318	2.323390	2.016542	7.975509	3.033511	3.468595
VMF L_3	3.164229	2.844004	2.546021	2.171431	9.949316	2.795862	3.911811
VMF L_2	2.672181	2.412301	2.120271	1.826469	8.389109	2.253622	3.278992
VMF L_1	2.539734	2.282574	2.027110	1.801360	8.022183	2.332868	3.167638
VMF L_2^2	3.120983	2.537216	2.346933	2.012454	8.366703	2.713251	3.516257
TNC_1-1*it*	2.793765	2.474302	2.213258	1.927886	8.430427	2.158107	3.332958
TNC_1-2*it*	2.881448	2.487709	2.247602	1.986638	8.502172	2.159408	3.377496
TNC_2-1*it*	2.698622	2.396683	2.150753	1.850226	8.221930	2.402061	3.286713
TNC_2-2*it*	2.748054	2.415496	2.183221	1.894070	8.277498	2.397230	3.319261
BVDF	2.503371	2.159451	1.985131	1.805755	7.702407	2.263618	**3.069956**
GVDF-n	2.567071	2.208090	2.025193	1.858734	7.790386	2.257199	3.117779
GVDF-$\frac{n^2}{2}$	2.783496	2.393211	2.164308	2.010955	8.294628	2.252612	3.316535
$DDF_{10\%}$	1.981587	2.496231	2.162296	1.802862	7.732706	2.262450	3.073022
$DDF_{25\%}$	1.984216	2.495018	2.173008	1.803224	7.773269	2.260552	3.081548
$DDF_{50\%}$	1.990134	2.497560	2.189269	1.803707	7.814217	2.270120	3.094168
$DDF_{75\%}$	1.995986	2.500340	2.210545	1.804405	7.865193	2.289583	3.111009
$DDF_{90\%}$	2.007212	2.510397	2.233837	1.803365	7.927858	2.304767	3.131240

Table B.4b: MCRE with uncorrelated Gaussian noise, $\sigma^2 = 1000$

Method	lily	autumn	boats	lenna	baboon	Sample1	average
Input	17.76449	20.04216	16.09797	12.11596	17.78781	6.533786	15.05703
Mean	4.910964	4.397273	4.317309	3.320604	8.769375	4.036637	**4.958694**
VMF L_3	17.36356	19.22668	15.38037	11.77335	19.36282	7.918022	15.17080
VMF L_2	9.236990	9.008574	8.325689	5.403453	11.64155	4.553740	8.028332
VMF L_1	8.072819	8.104848	7.224551	5.036543	10.90624	4.489202	7.305701
VMF L_2^2	7.817103	7.878883	7.037582	5.028633	10.88454	4.925498	7.262040
TNC_1-1*it*	10.15338	9.862814	9.450676	5.681638	12.40891	4.521704	8.679854
TNC_1-2*it*	10.44485	10.08679	9.640513	5.702325	12.60487	4.397200	8.812757
TNC_2-1*it*	8.768168	8.390999	7.737244	5.191567	11.29379	4.685376	7.677858
TNC_2-2*it*	8.962819	8.722909	8.128385	5.236946	11.58124	4.637244	7.878257
BVDF	6.433436	6.183230	5.740853	4.179840	9.685510	3.562559	5.964238
GVDF-n	8.134924	8.295863	7.332948	5.517409	10.94543	4.193970	7.403425
GVDF-$\frac{n^2}{2}$	11.86466	12.40715	10.40062	7.933870	13.97178	5.515068	10.34886
$DDF_{10\%}$	6.708487	7.279976	6.854556	4.360421	10.18673	3.902252	6.548738
$DDF_{25\%}$	7.181670	7.498918	7.135951	4.523824	10.40818	4.217890	6.827739
$DDF_{50\%}$	7.220765	7.826807	7.417644	4.695163	10.63405	4.295434	7.014977
$DDF_{75\%}$	7.606607	7.931087	7.834774	4.826300	10.79804	4.403762	7.233429
$DDF_{90\%}$	7.685904	8.203390	8.138985	4.932149	11.00700	4.513201	7.413438

Table B.4c: MCRE with partly correlated Gaussian noise, $\sigma^2 = 1000$

Method	lily	autumn	boats	lenna	baboon	Sample1	average
Input	15.21440	17.25814	13.42864	9.802049	15.11093	4.997064	12.63520
Mean	4.267031	3.643708	3.778935	2.940166	8.496480	3.846191	**4.495418**
VMF L_3	13.51280	14.51682	11.96799	8.776376	16.16197	6.591844	11.92130
VMF L_2	7.972579	7.800164	7.460981	4.767545	10.92195	4.189867	7.185515
VMF L_1	6.967286	6.778950	6.425772	4.361769	10.20310	4.073794	6.468445
VMF L_2^2	6.852767	6.678017	6.181173	4.462616	10.19165	4.544165	6.485065
TNC_1-1*it*	9.364194	8.659807	9.295141	4.833942	11.81300	3.843623	7.968285
TNC_1-2*it*	10.28187	9.199260	10.31617	4.856541	12.37177	3.609042	8.439108
TNC_2-1*it*	7.525032	7.172767	7.083144	4.451310	10.57692	4.176362	6.830922
TNC_2-2*it*	8.177134	7.469680	7.975916	4.458825	10.98549	4.017840	7.180814
continued							

Method	lily	autumn	boats	lenna	baboon	Sample1	average
BVDF	5.289128	4.902907	4.820331	3.548727	9.078407	3.051415	5.115152
GVDF-n	6.318058	6.125024	5.724906	4.514297	9.848216	3.294654	5.970859
GVDF-$\frac{n^2}{2}$	9.039535	9.178793	8.079015	6.287802	12.11291	4.474539	8.195433
$DDF_{10\%}$	5.888462	6.093749	5.668196	3.713106	9.559642	3.387029	5.718364
$DDF_{25\%}$	6.173932	6.369327	6.008336	3.864679	9.754896	3.606587	5.962959
$DDF_{50\%}$	6.437336	6.644820	6.314880	4.008862	9.951253	3.795938	6.192181
$DDF_{75\%}$	6.689202	6.922864	6.630039	4.145181	10.14484	3.963692	6.415969
$DDF_{90\%}$	6.890971	7.152767	6.896794	4.265980	10.31258	4.089701	6.601466

Table B.4d: MCRE with uncorrelated impulsive noise,$\rho = 10\%$

Method	lily	autumn	boats	lenna	baboon	Sample1	average
Input	9.961675	10.40569	9.653702	9.259682	10.41942	7.689333	9.564917
Mean	6.309910	5.819361	5.453885	4.519783	9.691044	4.999631	6.132269
VMF L_3	5.726241	4.797271	4.437855	2.997122	12.62391	3.632961	5.702560
VMF L_2	2.888138	2.523212	2.251151	1.916486	8.583076	2.325367	3.414572
VMF L_1	2.782383	2.415430	2.170042	1.921989	8.236787	2.474297	3.333488
VMF L_2^2	3.809764	3.173439	2.938953	2.818423	9.294507	3.256008	4.215182
TNC_1-1*it*	2.911558	2.543719	2.283848	1.970334	8.651369	2.245057	3.434314
TNC_1-2*it*	2.948321	2.544031	2.312232	1.999579	8.702102	2.247111	3.458896
TNC_2-1*it*	3.102298	2.650537	2.513357	2.149785	8.568698	2.629209	3.602314
TNC_2-2*it*	3.058256	2.639194	2.498090	2.138819	8.604372	2.557589	3.582720
BVDF	2.722601	2.247272	2.100353	1.921016	7.884865	2.390784	**3.211148**
GVDF-n	2.810622	2.318950	2.160865	1.999847	8.051425	2.382031	3.287290
GVDF-$\frac{n^2}{2}$	3.053193	2.592941	2.384045	2.186144	8.929833	2.385246	3.588567
$DDF_{10\%}$	2.089295	2.708027	2.252791	1.912736	7.912043	2.394981	3.211645
$DDF_{25\%}$	2.090403	2.700083	2.272189	1.911492	7.956760	2.392933	3.220643
$DDF_{50\%}$	2.100686	2.698935	2.290624	1.913098	8.012920	2.404119	3.236731
$DDF_{75\%}$	2.113657	2.712212	2.326475	1.914409	8.080936	2.426920	3.262435
$DDF_{90\%}$	2.135512	2.737155	2.358090	1.917890	8.150027	2.445676	3.290725

Table B.4e: MCRE with partly correlated impulsive noise,$\rho = 10\%$

Method	lily	autumn	boats	lenna	baboon	Sample1	average
Input	4.876950	4.922984	4.792735	4.849250	5.105083	4.491860	4.839810
Mean	5.190216	4.634020	4.491305	3.703397	9.041490	4.272988	5.222236
VMF L_3	4.360385	3.568806	3.221665	2.670181	11.28204	3.384886	4.747995
VMF L_2	2.757421	2.464662	2.170436	1.879066	8.477252	2.291483	3.340053
VMF L_1	2.633152	2.327692	2.081238	1.849694	8.094825	2.399192	3.230966
VMF L_2^2	3.609557	3.032281	2.804137	2.634851	8.974161	3.032781	4.014628
TNC_1-1*it*	2.802045	2.478269	2.200388	1.918530	8.472594	2.192926	3.344126
TNC_1-2*it*	2.844498	2.487055	2.224665	1.948047	8.524417	2.192228	3.370152
TNC_2-1*it*	2.930838	2.529054	2.306536	2.042230	8.383184	2.505061	3.449484
TNC_2-2*it*	2.916129	2.506456	2.295888	2.036555	8.393293	2.444321	3.432107
BVDF	2.593666	2.187090	2.028865	1.844666	7.756820	2.333280	**3.124064**
GVDF-n	2.656632	2.237263	2.069519	1.905965	7.865341	2.334002	3.178121
GVDF-$\frac{n^2}{2}$	2.880089	2.441742	2.233910	2.071394	8.467205	2.327479	3.403637
$DDF_{10\%}$	2.019750	2.580495	2.191116	1.841219	7.792552	2.342006	3.127856
$DDF_{25\%}$	2.021879	2.578327	2.204213	1.841265	7.833236	2.345214	3.137356
$DDF_{50\%}$	2.026446	2.579392	2.225423	1.842641	7.876841	2.352019	3.150460
$DDF_{75\%}$	2.036566	2.586871	2.243645	1.844141	7.931920	2.355918	3.166510
$DDF_{90\%}$	2.050386	2.595914	2.273234	1.846972	8.002157	2.381169	3.191639

Table B.4f: MCRE with uncorrelated mixed noise, $\sigma^2 = 1000$ and $\rho = 10\%$

Method	lily	autumn	boats	lenna	baboon	Sample1	average
Input	23.91495	26.16056	22.12074	18.72687	24.40714	12.78261	21.35214
Mean	6.870906	6.580797	6.204120	5.263344	10.26542	5.583407	**6.794666**
VMF L_3	19.46994	20.65776	17.24895	13.13758	21.76520	8.521240	16.80011
VMF L_2	10.35751	10.21145	9.115226	5.926112	12.49455	4.819130	8.820665
VMF L_1	9.042378	8.964632	7.994265	5.745807	11.68404	4.919518	8.058439
VMF L_2^2	9.195767	9.105600	8.343601	6.512791	12.09517	6.280282	8.588869
TNC_1-1*it*	11.73230	11.46484	10.40887	6.275385	13.49947	4.712200	9.682178
TNC_1-2*it*	11.68319	11.82950	10.74583	6.246431	13.84586	4.657756	9.834763
TNC_2-1*it*	9.875655	9.818248	8.762564	6.056860	12.36967	5.485846	8.728141
TNC_2-2*it*	10.35896	10.53569	9.3210656	6.052041	12.72796	5.518137	9.085641
continued							

Method	lily	autumn	boats	lenna	baboon	Sample1	average
BVDF	7.466984	7.235387	6.749495	4.910563	10.47866	3.953588	6.799113
GVDF-n	9.500832	9.690325	8.373263	6.406437	11.96332	4.491467	8.404274
GVDF-$\frac{n^2}{2}$	13.69771	14.48971	11.81162	9.279344	15.70924	6.024821	11.83541
$DDF_{10\%}$	7.8677154	8.361334	8.068290	5.034047	11.10221	4.369034	7.467104
$DDF_{25\%}$	8.0009423	8.630889	8.279885	5.271369	11.24918	4.618434	7.675117
$DDF_{50\%}$	8.2533982	8.911952	8.691646	5.409336	11.50967	4.846235	7.937039
$DDF_{75\%}$	8.487206	9.241735	9.009858	5.598140	11.77407	4.887219	8.166371
$DDF_{90\%}$	8.658783	9.378723	9.170963	5.691297	11.93358	5.031731	8.310846

Table B.4g: MCRE with partly correlated mixed noise, $\sigma^2 = 1000$ and $\rho = 10\%$

Method	lily	autumn	boats	lenna	baboon	Sample1	average
Input	18.70517	20.64950	17.01012	13.76564	18.86368	9.035726	16.33831
Mean	5.674435	5.301962	5.213688	4.344940	9.480184	4.870401	5.814268
VMF L_3	14.78553	15.19625	12.95473	9.852570	17.77565	7.291724	12.97608
VMF L_2	8.386130	8.221307	7.773000	4.993254	11.21906	4.321035	7.485631
VMF L_1	7.150395	6.960154	6.597192	4.560898	10.37484	4.246282	6.648294
VMF L_2^2	7.442313	7.386486	6.803903	5.308061	10.81191	5.296592	7.174877
TNC_1-1*it*	9.646336	8.974909	9.231364	5.024119	11.97574	4.011453	8.143988
TNC_1-2*it*	10.28534	9.405073	10.01837	5.039900	12.42863	3.839282	8.502765
TNC_2-1*it*	7.535721	7.318361	6.955509	4.724423	10.70582	4.446031	6.947644
TNC_2-2*it*	8.116209	7.658124	7.651891	4.736758	11.07793	4.312507	7.258902
BVDF	5.541826	5.156657	5.044568	3.749016	9.282217	3.183603	**5.326314**
GVDF-n	6.575810	6.393166	5.961454	4.742968	10.07236	3.430074	6.195973
GVDF-$\frac{n^2}{2}$	9.475436	9.753138	8.407260	6.637653	12.54824	4.669118	8.581807
$DDF_{10\%}$	6.156775	6.410440	6.009956	3.921614	9.840348	3.625043	5.994029
$DDF_{25\%}$	6.433951	6.679972	6.339751	4.076480	10.03784	3.873293	6.240215
$DDF_{50\%}$	6.692600	6.949436	6.662580	4.228637	10.23986	4.067426	6.473423
$DDF_{75\%}$	6.935237	7.225021	6.966647	4.374687	10.43921	4.234906	6.695952
$DDF_{90\%}$	7.167052	7.446084	7.226991	4.491951	10.61363	4.362562	6.884712

Appendix C

Watershed Evaluation

These are the results for section 7.3 on watershed gradient processing. The first stage of evaluation was in region numbers and NMSE for all mask-based methods and gradient related sieves (VAMS, VDMS), on the standard 6 image set of appendix D.1. Results include average MSE, region numbers, and metrics for 'lily', 'lenna'. For the greyscale and colour comparison, the MSE is between the input and region-averaged colour images, taking the L_2 magnitude of the gradient. This MSE against the basin-averaged image gives a measure of the mismatch between input and the identified regions, with inhomogeneous groupings getting worse results for a given number of regions; in essence, this is an homogeneity measure.

The second stage, considering gradient processing, was post-processing and segmentation, done with the Berkeley database. 40 images were taken, and the combined L_2 colour or greyscale gradient was sieved with contrast, volume, power and area. Since the database takes a single boundary map and the watershed boundaries become unstable when basin boundaries thicken, the results were trimmed to fit lower level segmentations; an alternative for greyscale (and possibly colour) gradients would be dynamics of contours [46]. This had a visible impact on closings, as seen in figures C.1b-c. Here, the process removes most of the random watershed boundaries at higher (darker) scales, visibly improving segmentation quality for closings while hardly affecting the combined opening and closing in figures C.1e-f.

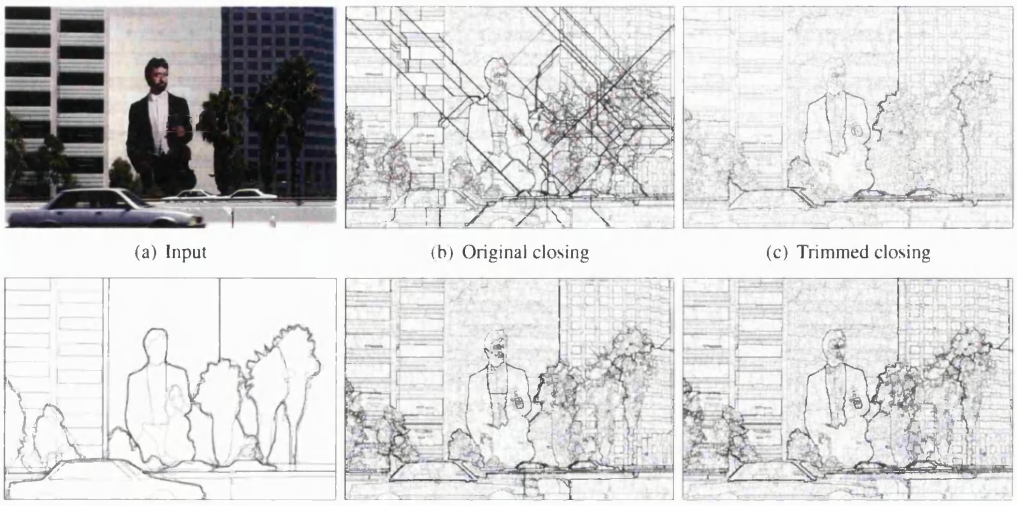
Trimming, as used here, is a way to remove the random watershed borders (often at $0°, 45°, \ldots$) when not enough edges exist in the watershed gradient. Since removed boundaries do not usually reappear or are only partly reconstructed from segments, any new boundary points which do not fall within 1 pixel of currently existing points are removed. This gives a single edge map with scale-space causality, which is easily processed by the Berkeley database. Another alternative is to remove only edges not present at the initial scale, keeping most of the closed boundaries, yet with a cost to segmentation performance: the area closing gives an overall segmentation score of FoM=0.50, whereas the trimmed closing gives FoM=0.52.

Method	Scale	Average		lily		lenna	
		MSE	Regions	MSE	Regions	MSE	Regions
Input		141.5281	11430	117.4714	3303	42.4950	22700
	1	152.1896	8623	143.2175	2697	48.4546	21074
VAMCS	2	187.3342	5898	197.2890	1679	61.1402	14025
L_2	5	259.1544	2829	306.4392	835	85.8521	6447
gradient	10	324.0038	1453	404.3163	472	112.8037	3226
	50	566.5444	283	1087.0400	121	202.9900	580
	100	754.1647	144	1569.4861	47	259.4464	263
	2	221.9420	4285	237.5404	1538	74.9932	10085
VAMOCS	5	325.9923	1845	457.9393	711	109.0762	4275
L_2	10	423.2354	964	719.7622	376	150.5766	2209
gradient	50	740.9516	221	1606.4734	94	262.1599	478
	100	967.0326	120	2254.2464	56	375.3619	307
	1	154.6460	9673	147.6819	2776	48.8130	21085
VDMCS	2	242.1503	4389	271.3350	1278	79.8617	9504
L_2	5	348.6985	1598	424.6908	529	124.4924	3472
gradient	10	465.8990	744	671.4023	259	178.7562	1598
	50	888.4283	134	1705.5398	51	348.8325	266
	100	1163.6661	66	2557.1350	25	477.4242	119
	2	247.2858	4166	271.6468	1296	81.8500	8934
VDMOCS	5	364.7923	1468	543.3016	513	129.7490	3127
L_2	10	478.9340	720	820.9953	265	179.0905	1480
gradient	50	846.6861	149	1846.0712	59	353.7077	282
	100	1074.8981	74	2262.0954	35	459.8619	146
Mean	5*5	298.6237	6689	356.5066	1743	97.9657	12386
	13*13	549.8884	4580	1032.1465	1031	226.2310	7332
VMF, L_1	5*5	218.0111	7619	222.4377	2409	62.0094	17260
	13*13	368.2743	5592	546.5521	2290	104.0318	14662
TVMF, L_1	5*5	220.8731	8040	214.3260	2661	56.5687	18498
	13*13	370.5733	5872	545.6031	2385	104.3118	15483

Table C.1: Watershed for colour processing methods

Method	Scale	Average		lily		lenna	
		MSE	Regions	MSE	Regions	MSE	Regions
Input		141.5281	11430	117.4714	3303	42.4950	22700
Y-AOC	10	176.409	10382	277.8222	3482	46.3701	21874
Y-AOC	100	281.525	6381	484.7221	2046	77.5916	15143
SE-ASF	3*3	112.630	10309	111.6217	2842	34.2835	20448
SE-ASF	9*9	168.099	9540	240.6648	2732	55.2211	18273
Mean	5*5	228.0477	6843	296.9547	1801	78.3551	12613
Mean	13*13	433.6824	4710	887.5302	1063	189.4042	7441
Median	5*5	163.2952	6790	182.3050	2098	47.7569	13745
Median	13*13	308.7665	4430	556.9156	1503	96.9587	8865

Table C.2: Watershed for greyscale processing methods



(a) Input (b) Original closing (c) Trimmed closing

(d) Human segmentation (e) Original sieving (f) Trimmed sieving

Figure C.1: Effect of trimming on multi-scale watershed of area sieves

Attribute	Sieve	FoM(R,P)	Attribute value
Volume	untrimmed	0.5102(0.7028,0.4004)	2^{15}
Power	greyscale	**0.5138**(0.6590,0.4214)	2^{18}
Area	closing	0.4967(0.5917,0.4280)	40000
Contrast	greyscale	0.4641(0.5822,0.3858)	60
Volume	AOC	**0.5383**(0.6403,0.4643)	2^{15}
Power		0.5316(0.6254,0.4623)	2^{20}
Area		0.5171(0.5820,0.4653)	400
Contrast	greyscale	0.4286(0.5405,0.3550)	40
Volume	closing	**0.5455**(0.5954,0.5033)	2^{15}
Power		0.5343(0.7095,0.4285)	2^{20}
Area		0.5255(0.5720,0.4859)	200
Contrast	PCS	0.4887(0.5720,0.4266)	120
Volume	opening	0.5352(0.5734,0.5018)	2^{18}
Power		**0.5388**(0.6460,0.4621)	2^{24}
Area		0.5134(0.6544,0.4224)	6000
Contrast	colour	0.4991(0.6571,0.4023)	140
Volume	AOC	0.5263(0.6287,0.4526)	2^{21}
Power		**0.5366**(0.6652,0.4498)	2^{26}
Area		0.5006(0.7147,0.3852)	800
Contrast	colour	0.4535(0.7146,0.3321)	100
Volume	closing	0.5368(0.6675,0.4488)	2^{21}
Power		**0.5386**(0.7189,0.4307)	2^{22}
Area		0.5172(0.6146,0.4465)	600

Table C.3: Watershed for gradient sieves

Appendix D

Extra images

D.1 Sieving results

(a) autumn (b) lily (c) sample1

(d) lenna (e) boats (f) baboon

Figure D.1: Test images

(a) VAMOCS sieving to 200/765 (b) VAMOCS sieving to 400/765 (c) VAMOCS sieving to 600/765

(d) VAMOCS sieving to 100/442 (e) VAMOCS sieving to 200/442 (f) VAMOCS sieving to 300/442

Figure D.2: Line contrast sieving

(a) VAMS sieving to 50/442 (b) VAMOCS sieving to 50/442 (c) CCS sieving to 50/442

(d) VAMS sieving to 100/442 (e) VAMOCS sieving to 100/442 (f) CCS sieving to 100/442

Figure D.3: Line contrast sieving(cont)

(a) VAMS sieve to 90/360deg (b) VAMOCS sieve to 90/360deg (c) CCS sieve to 90/360deg

(d) VAMS sieve to 180/360deg (e) VAMOCS sieve to 180/360deg (f) CCS sieve to 180/360deg

(g) VAMS sieve to 270/360deg (h) VAMOCS sieve to 270/360deg (i) CCS sieve to 270/360deg

(j) VAMS sieve to 350/360deg (k) VAMOCS sieve to 350/360deg (l) CCS sieve to 360/360deg

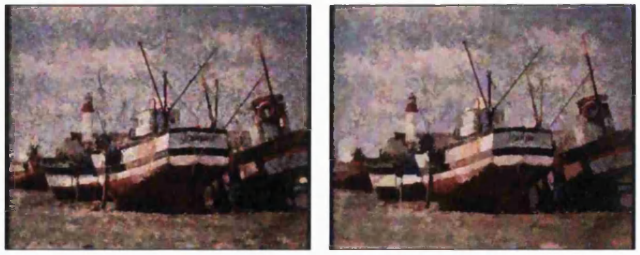
Figure D.4: Hue contrast sieving(cont)

(a) corrupted input, $\rho = 10\%$ (b) VAMOCS to area 10 (c) sieve to $500*10^3$, clean input

(d) sieve to $10*10^3$ (e) sieve to $150*10^3$ (f) sieve to $300*10^3$

Figure D.5: VAMOCS power sieve vs impulsive noise

(a) input sieve to $5*10^5$, 3085 reg (b) corrupted input, $\sigma^2 = 1000$, 103618 reg (c) sieve to 10^4, 33114 reg



(d) sieve to 10^5, 7555 reg (e) sieve to $5*10^5$, 2027 reg

Figure D.6: VAMS power sieve vs Gaussian noise

(a) input at $5*10^5$, 1591 reg

(b) corrupted input, $\sigma^2 = 1000$, 71061 reg

(c) sieve to 10^4, 20480 reg

(d) sieve to 10^5, 4991 reg

(e) sieve to $5*10^5$, 1774 reg

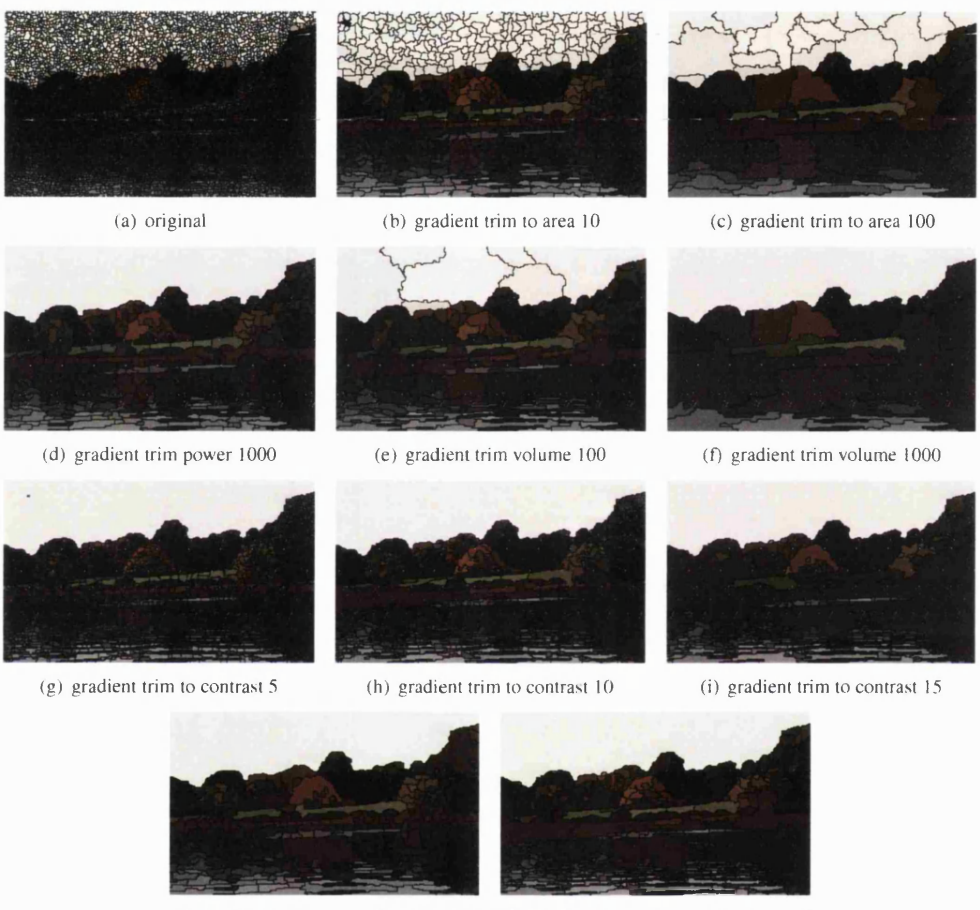
Figure D.7: VAMOCS power sieve vs Gaussian noise

(a) original

(b) gradient trim to area 10

(c) gradient trim to power 1000

Figure D.8: Example watershed with attribute sieving



(a) original (b) gradient trim to area 10 (c) gradient trim to area 100

(d) gradient trim power 1000 (e) gradient trim volume 100 (f) gradient trim volume 1000

(g) gradient trim to contrast 5 (h) gradient trim to contrast 10 (i) gradient trim to contrast 15

(j) gradient trim to area 10+contrast 3 (k) gradient trim power 1000+contrast 3

Figure D.9: Example watershed with attribute sieving(cont)

(a) inputs

(b) VAMS, area 100

(c) CCS, area 100

(d) VAMS, area 10000

(e) CCS, area 10000

(f) VAMOCS, area 100

(g) VDMOCS, area 100

(h) VAMOCS, area 10000

(i) VDMOCS, area 10000

(j) GSAOC, area 100 (k) PCS-M, area 100

(l) GSAOC, area 10000 (m) PCS-M, area 10000

(n) PCS-V, area 100 (o) PCS-A, area 100

(p) PCS-V, area 10000 (q) PCS-A, area 10000

(r) PCOCS-V, area 100 (s) PCOCS-A, area 100

(t) PCOCS-V, area 10000 (u) PCOCS-A, area 10000

Figure D.10: Sieving for original(L) and corrupted 'lenna'(R) in L_{ae2} metric

(a) inputs

(b) VAMS, area 100

(c) CCS, area 100

(d) VAMS, area 10000

(e) CCS, area 10000

(f) VAMOCS, area 100

(g) VDMOCS, area 100

(h) VAMOCS, area 10000

(i) VDMOCS, area 10000

(j) GSAOC, area 100 (k) PCS-M, area 100

(l) GSAOC, area 10000 (m) PCS-M, area 10000

(n) PCS-V, area 100 (o) PCS-A, area 100

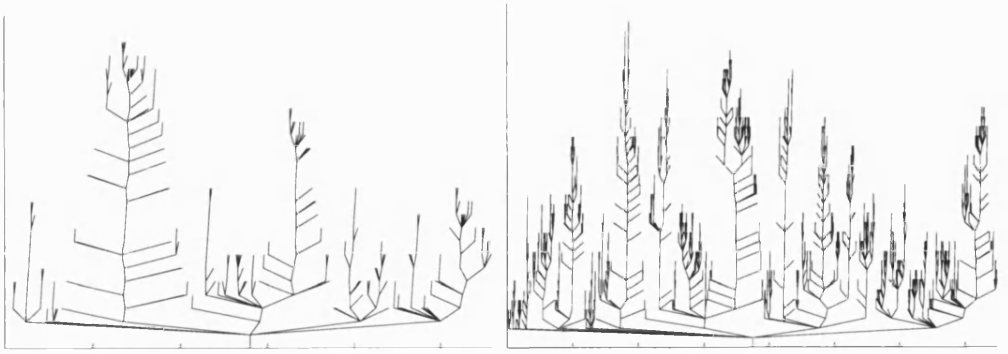
(p) PCS-V, area 10000 (q) PCS-A, area 10000

(r) PCOCS-V, area 100 (s) PCOCS-A, area 100

(t) PCOCS-V, area 10000 (u) PCOCS-A, area 10000

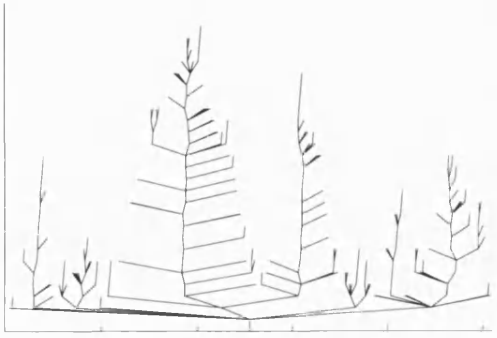
Figure D.11: Sieving for original(L) and corrupted 'lenna'(R) images for L_2 metric

D.2 Tree results

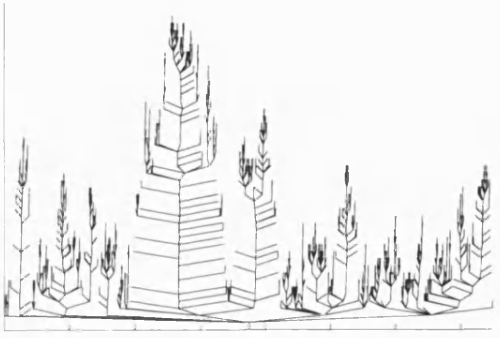


(a) Area,279 regions

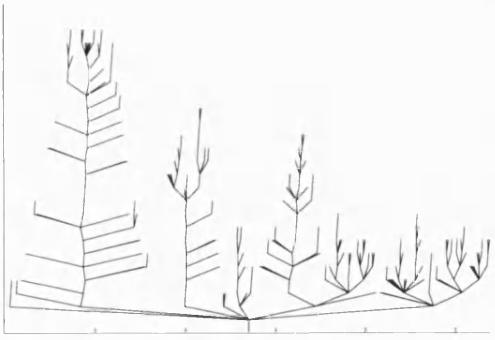
(b) Area,1500 regions



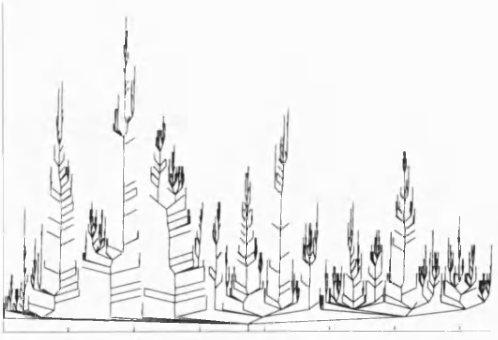
(c) Contrast,252 regions



(d) Contrast,1500 regions

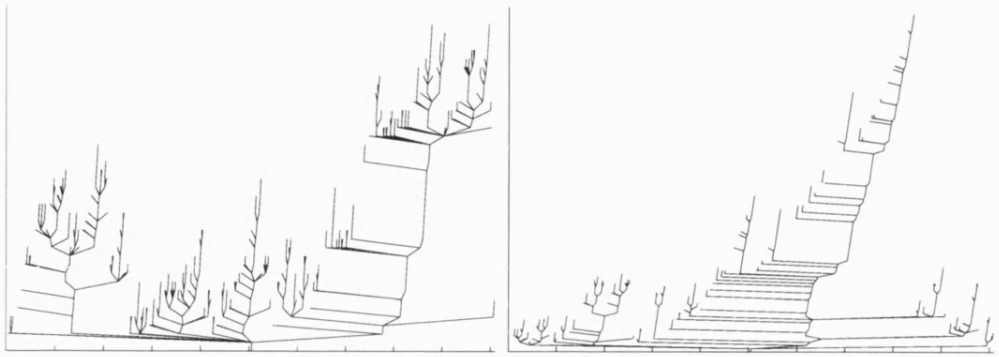


(e) Power,269 regions

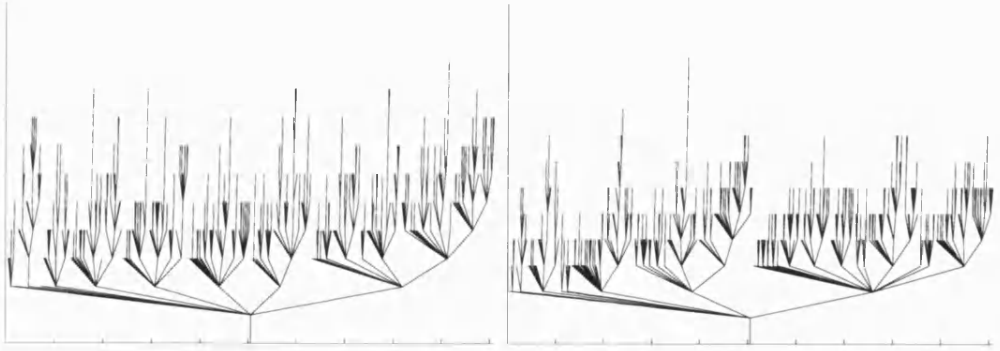


(f) Power,1500 regions

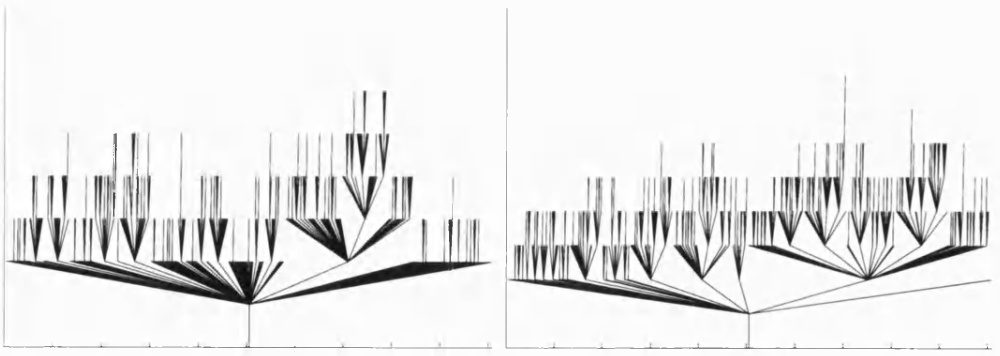
Figure D.12: Effects of attribute on tree structure on 'lenna', GSAOC



(a) VAMS (b) VDMS

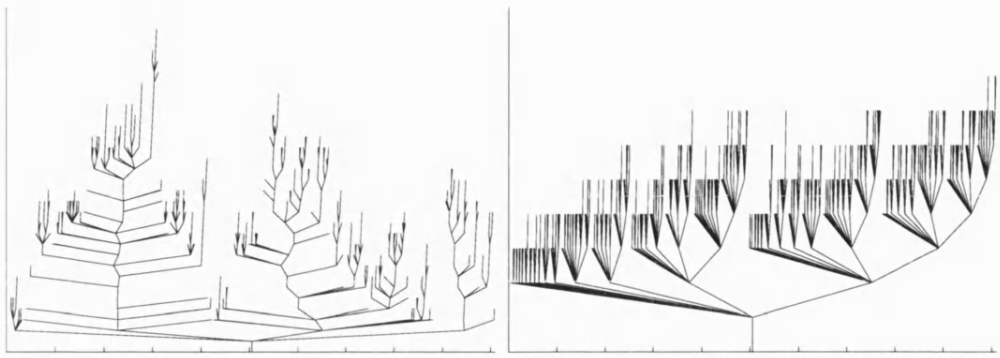


(c) VAMOCS (d) VDMOCS

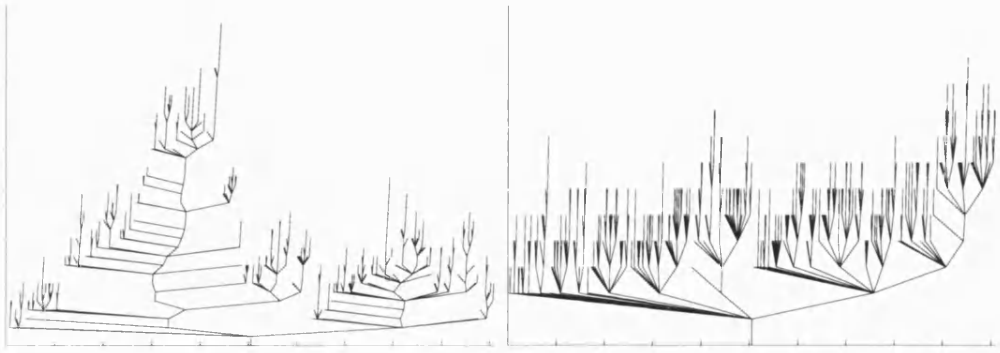


(e) VAMS-close (f) VDMS-close

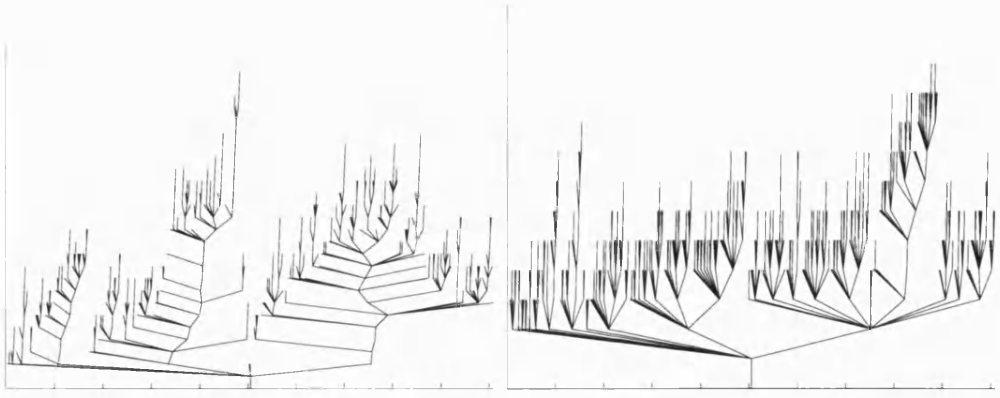
Figure D.13: Sieve comparison at 500 regions for 'lily'



(a) GSAOC (b) CCS



(c) PCSA (d) PCOCSA



(e) PCSV (f) PCOCSV

Figure D.14: Sieve comparison at 500 regions for 'lily' (cont)

References

[1] The Wikipedia. Wikimedia Foundation, Inc., 2006. http://en.wikipedia.org/wiki/.

[2] Robert Leggat. A History of photography, 1995. http://www.rleggat.com/photohistory/index.html.

[3] LetgoDigital.com, April 2006. http://www.letsgodigital.org/en/news/articles/story_6266.html, February 2006; http://www.letsgodigital.org/en/news/articles/story_3018.html, March 2005.

[4] M.C. d'Ornellas. A multi-scale gradient approach for color-based morphological segmentation. In *Int. Conf. on Pattern Recognition*, volume 3, pages 363–366, September 2000.

[5] P. Lambert and H. Grecu. A quick and coarse color image segmentation. In *IEEE Int. Conf. on Image Processing*, pages I: 965–968, 2003.

[6] Pascal Vuylsketer. Computer Graphics: Color Topics in Computer Graphics. Australian National University, April 2004. http://cs.anu.edu.au/escience/lecture/cg/Color/-HSV_HLS_MostCommUsed.en.html.

[7] A.G. Hanbury and J. Serra. Morphological operators on the unit circle. *IEEE Trans. Image Processing*, 10(12):1842–1850, December 2001.

[8] A. Shiji and N. Hamada. Color image segmentation method using watershed algorithm and contour information. In *IEEE Int. Conf. on Image Processing*, volume 4, pages 305–309, October 1999.

[9] C. Zhang and P. Wang. A new method of color image segmentation based on intensity and hue clustering. In *Int. Conf. on Pattern Recognition*, volume 3, pages 613–616, 2000.

[10] J. Angulo and J. Serra. Color segmentation by ordered mergings. In *IEEE Int. Conf. on Image Processing*, volume 2, pages 125–128, September 2003.

[11] Colour Space FAQ, April 2004. http://www.ilkeratalay.com/colorspacesfaq.php.

[12] K.N. Plataniotis and A.N. Venetsanopoulos. *Color Image Processing and Applications*. Springer-Verlag, August 2000.

[13] L. Shafarenko, M. Petrou, and J.V. Kittler. Histogram based segmentation in a perceptually uniform color space. *IEEE Trans. Image Processing*, 7(9):1354–1358, September 1998.

[14] Moon Gi. Kang. *Selected papers on CCD and CMOS imagers.* SPIE milestone series ; v. MS177. Bellingham, Wash. : SPIE, 2003. 636pp.

[15] Concepts in Digital Imaging Technology. Olympus America Inc., Florida State University, 2004. http://micro.magnet.fsu.edu/primer/digitalimaging/concepts/concepts.html.

[16] T. Viero, K. Oistamo, and Y. Neuvo. Three-dimensional median-related filters for color image sequence filtering. *IEEE Trans. Circuits and Systems for Video Technology*, 4(2):129–142, 208–10, April 1994.

[17] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int. Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[18] S. Albin, G. Rougeron, B. Peroche, and A. Tremeau. Quality image metrics for synthetic images based on perceptual color differences. *IEEE Trans. Image Processing*, 11(9):961–971, September 2002.

[19] P.E. Trahanias and A.N. Venetsanopoulos. Vector directional filters: A new class of multichannel image processing filters. *IEEE Trans. Image Processing*, 2(4):528–534, October 1993.

[20] D.R. Charles. *Algorithmic and learning based filtering techniques with application to colour image noise suppression and enhancement.* Phd thesis, Machine Vision Group, Department of Physics, Royal Holloway, University of London, 2003.

[21] A.J. Bardos and S.J. Sangwine. Image distortions produced by mean, median and mode filters. *IEE Colloquium on Non-Linear Signal and Image Processing*, pages 8/1–8/4, May 1998.

[22] L. Lucchese and S.K. Mitra. Color image segmentation: A state-of-the-art survey (invited paper). In *Image Processing, Vision, and Pattern Recognition, Proc. of the Indian National Science Academy (INSA-A), New Delhi, India*, volume 67A(2), pages 207–221, March 2001.

[23] L. Heucke, M. Knaak, and R. Orglmeister. A new image segmentation method based on human brightness perception and foveal adaptation. *IEEE Signal Processing Letters*, 7(6):129–131, June 2000.

[24] N. Fletcher and A.N. Evans. Minimum distance texture classification of sar images using wavelet packets. In *IEEE Int. Geoscience and Remote Sensing Symposium*, volume 3, pages 1438–1440, June 2002.

[25] T. Allen and T.L. Huntsberger. Comparing color edge detection and segmentation methods. In *Proceedings of Southeastcon*, pages 722–728, 1989.

[26] S. Wesolkowski, R.D. Dony, and M.E. Jernigan. Global color image segmentation strategies: Euclidean distance vs. vector angle. In *Proc. IEEE Signal Processing Society Workshop on Neural Networks for Signal Processing IX*, pages 419–428, 1999.

[27] J. Serra. Image segmentation. In *IEEE Int. Conf. on Image Processing*, volume 1, pages 345–348, September 2003.

[28] M.A. Ruzon and C. Tomasi. Color edge detection with the compass operator. In *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 160–166, 1999.

[29] D. Zugaj and V. Lattuati. A new approach of color images segmentation based on fusing region and edge segmentations outputs. *Pattern Recognition*, 31(2):105–113, February 1998.

[30] IP Operator Worksheets, October 2003. http://homepages.inf.ed.ac.uk/rbf/HIPR2/wksheets.htm.

[31] T. Leung and J. Malik. Contour continuity in region-based image segmentation. In *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, volume 1, pages 544–559, June 1998.

[32] S.C. Zhu, T.S. Lee, and A.L. Yuille. Region competition: unifying snakes, region growing, energy/bayesian/MDL for multi-band image segmentation. In *IEEE Int. Conf. on Computer Vision*, pages 416–423, June 1995.

[33] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *IEEE Int. Conf. on Computer Vision*, pages 694–699, June 1995.

[34] Ramani Pichumani. Snakes: an active model. CVonline, 1997. http://homepages.inf.ed.ac.uk/-rbf/CVonline/LOCAL_COPIES/RAMANI1/node31.html.

[35] Hieu Tat Nguyen, M. Worring, and R. van den Boomgaard. Watersnakes: energy-driven watershed segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(3):330–342, 2003.

[36] E. Navon, O. Miller, and A. Averbuch. Color image segmentation based on automatic derivation of local thresholds. In C. Sun, H. Talbot, S. Ourselin, and T. Adriaansen, editors, *Proc. VIIth Digital Image Computing: Techniques and Applications*, pages 571–580, December 2003. Sidney.

[37] F.H.Y. Chan, F.K. Lam, and H. Zhu. Adaptive thresholding by variational method. *IEEE Trans. Image Processing*, 7(3):468–473, March 1998.

[38] T. Carron and P. Lambert. Color edge detector using jointly hue, saturation, and intensity. In *IEEE Int. Conf. on Image Processing*, volume 3, pages 977–981, November 1994.

[39] T. Lindeberg. Scale-space theory: A basic tool for analysing structures at different scales. *AppStat*, 21(2):224–270, 1994.

[40] A. Bosson, R.W. Harvey, and J.A. Bangham. Scale-space pattern processors: are they robust to noise and occlusion? In *IEE Colloquium on Pattern Recognition (Digest No. 1997/018)*, pages 8/1–8/6, February 1997.

[41] F. Cheng and A.N. Venetsanopoulos. An adaptive morphological filter for image processing. *IEEE Trans. Image Processing*, 1(4):533–539, October 1992.

[42] A. Wright and S. Acton. Watershed pyramids for edge detection. In *IEEE Int. Conf. on Image Processing*, volume 2, pages 578–581, October 1997.

[43] L. Shafarenko, M. Petrou, and J.V. Kittler. Automatic watershed segmentation of randomly textured color images. *IEEE Trans. Image Processing*, 6(11):1530–1544, November 1997.

[44] K. Saarinen. Color image segmentation by a watershed algorithm and region adjacency graph processing. In *IEEE Int. Conf. on Image Processing*, volume 3, pages 1021–1025, Nov 1994.

[45] R. Manduchi. Bayesian fusion of color and texture segmentations. In *IEEE Int. Conf. on Computer Vision*, volume 2, pages 956–962, September 1999.

[46] L. Najman and M. Schmitt. Geodesic saliency of watershed contours and hierarchical segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(12):1163–1173, December 1996.

[47] K. Haris, S.N. Efstratiadis, N. Maglaveras, and A.K. Katsaggelos. Hybrid image segmentation using watersheds and fast region merging. *IEEE Trans. Image Processing*, 7(12):1684–1699, December 1998.

[48] C.R. Jung and J. Scharcanski. Robust watershed segmentation using the wavelet transform. In *XV Brazilian Symposium on Computer Graphics, Image Processing*, pages 131–137, October 2002.

[49] V. Gies and T.M. Bernard. Statistical solution to watershed over-segmentation. In *IEEE Int. Conf. on Image Processing*, volume 3, pages 1863–1866, October 2004.

[50] J.B.T.M. Roerdink and A. Meijster. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta Informaticae*, 41:187–188, 2000.

[51] F.G. Meyer. Color image segmentation. In *IEEE Int. Conf. on Image Processing*, pages 303–306, May 1992.

[52] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(82):888–905, August 2000.

[53] E. Sharon, M. Galun, D. Sharon, R. Basri, and A. Brandt. Hierarchy and adaptivity in segmenting visual scenes. *Nature*, 442:810–813, August 2006. http://www.nature.com/nature-/journal/v442/n7104/full/nature04977.html.

[54] P. Soundararajan and S. Sarkar. An in-depth study of graph partitioning measures for perceptual organization. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(6):642–660, June 2003.

[55] D. Gatica-Perez, C. Gu, M.-T. Sun, and S. Ruiz-Correa. Extensive partition operators, gray-level connected operators, and region merging/classification segmentation algorithms: theoretical links. *IEEE Trans. Image Processing*, 10(9):1332–1345, September 2001.

[56] P. Salembier and L. Garrido. Binary partition tree as an efficient representation for image processing, segmentation and information retrieval. *IEEE Trans. Image Processing*, 9(4):561–576, April 2000.

[57] A. Gillet, L. Macaire, C. Botte-Lecocq, and J.G. Postaire. Color image segmentation based on fuzzy mathematical morphology. In *IEEE Int. Conf. on Image Processing*, volume 3, pages 348–351, 2000.

[58] F.C. Flores, R.Jr. Hirata, J. Barrera, R.A. Lotufo, and F. Meyer. Morphological operators for segmentation of color sequences. *XIII Brazilian Symposium on Computer Graphics and Image Processing*, pages 300–307, October 2000.

[59] P. Brigger, S. Ayer, and M. Kunt. Morphological shape representation of segmented images based on temporally modeled motion vectors. In *IEEE Int. Conf. on Image Processing*, volume 3, pages 756–760, November 1994.

[60] S. Mukhopadhyay and B. Chanda. Multiscale morphological segmentation of gray-scale images. *IEEE Trans. Image Processing*, 12(5):533–549, May 2003.

[61] T. Westman, D.A. Harwood, T. Laitinen, and M. Pietikainen. Color segmentation by hierarchical connected components analysis with image enhancements by symmetric neighborhood filters. In *ICPR90*, pages Vol–I 796–802, 1990.

[62] D. Gimenez and A.N. Evans. Colour morphological scale-spaces for image segmentation. In *Proc. British Machine Vision Conf.*, volume 2, pages 909–918, September 2005.

[63] D. Gimenez and A.N. Evans. Colour morphological scale-spaces from the positional colour sieve. In *Digital Image Computing: Techniques and Applications*, pages 415–422, December 2005.

[64] Heng-Da Cheng and Ying Sun. A hierarchical approach to color image segmentation using homogeneity. *IEEE Trans. Image Processing*, 9(12):2071–2082, December 2000.

[65] Wei Yu, J. Fritts, and Fangting Sun. A hierarchical image segmentation algorithm. In *IEEE Int. Conf. on Multimedia and Expo*, volume 2, pages 221–224, Aug 2002.

[66] S. Ji and H.W. Park. Image segmentation of color image based on region coherency. In *IEEE Int. Conf. on Image Processing*, volume 1, pages 80–83, October 1998.

[67] Y. Lan, J. R. Perez-Torres, and Harvey R. Finding stable salient contours. In *Proc. British Machine Vision Conf.*, volume 1, pages 30–39, September 2005.

[68] Chin-Hwa Kuo and Tay-Shen Wang. A real-time segmentation scheme for continuous color images. In *IEEE Int. Symposium on Circuits and Systems*, volume 2, pages 385–388, May 2001.

[69] E. Sifakis, I. Grinias, and G. Tziritas. Video segmentation using fast marching and region growing algorithms. *EURASIP Journal on Applied Signal Processing*, 4:379–388, April 2002.

[70] P. Villegas and X. Marichal. Perceptually-weighted evaluation criteria for segmentation masks in video sequences. *IEEE Trans. Image Processing*, 13(8):1092–1103, August 2004.

[71] Xu Jie and Shi Peng-fei. Natural color image segmentation. In *IEEE Int. Conf. on Image Processing*, volume 1, pages 973–976, September 2003.

[72] E. Saber, A.M. Tekalp, and G. Bozdagi. Fusion of color and edge information for improved segmentation and edge linking. In *IEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pages 2176–2179, 1996.

[73] Liu Rujie and Yuan Baozong. Unsupervised color image segmentation. In *Int. Conf. on Signal Processing*, volume 1, pages 744 -747, August 2002.

[74] O. Lezoray and A. Elmoataz. Graph based smoothing and segmentation of color images. In *Int. Symposium on Signal Processing and Its Applications*, volume 1, pages 517–520, July 2003.

[75] C. Fowlkes, D. Martin, and J. Malik. Learning affinity functions for image segmentation: Combining patch-based and gradient-based approaches. *Computer Vision and Pattern Recognition (CVPR)*, June 2003.

[76] T. Zoller, L. Hermes, and J.M. Buhmann. Combined color and texture segmentation by parametric distributional clustering. In *Int. Conf. on Pattern Recognition*, volume 2, pages 627–630, August 2002.

[77] S.D. Kamvar, D. Klein, and C.D. Manning. Interpreting and extending classical agglomerative clustering algorithms using a model-based approach. In *Int. Conf. on Machine Learning*, July 2002.

[78] E.J. Pauwels and G. Frederix. Cluster-based segmentation of natural scenes. In *IEEE Int. Conf. on Computer Vision*, pages 997–1002, 1999.

[79] L. Liu, Y. Dong, X. Song, and G. Fan. An entropy based segmentation algorithm for computer-generated document images. In *IEEE Int. Conf. on Image Processing*, volume 1, pages 541–544, September 2003.

[80] T. Geraud, P.Y. Strub, and J. Darbon. Color image segmentation based on automatic morphological clustering. In *IEEE Int. Conf. on Image Processing*, volume 3, pages 70–73, 2001.

[81] A. Wright and S. Acton. The fuzzy integral for color seal segmentation on document images. In *IEEE Int. Conf. on Image Processing*, volume 1, pages 157–160, September 2003.

[82] C. Connolly, S. Littlewood, and E.S. King. A system for the segmentation of colour images. In *IEEE Int. Conf. on Image Processing*, pages 441–444, July 1989.

[83] X. Munoz, J. Freixenet, X. Cufi, and J. Marti. Active regions for colour texture segmentation integrating region and boundary information. In *IEEE Int. Conf. on Image Processing*, volume 3, pages 453–456, September 2003.

[84] R.A. Reyna, N. Hernandez, D. Esteve, and M. Cattoen. Segmenting images with support vector machines. In *IEEE Int. Conf. on Image Processing*, volume 1, pages 820–823, September 2000.

[85] Mohamad H. Hassoun. *Fundamentals of artificial neural networks*. Cambridge, Mass. ; London : MIT Press, 1995.

[86] Robert L. Harvey. *Neural network principles*. London : Prentice-Hall International, 1994.

[87] Wang Lei and Qi Feihu. Adaptive fuzzy Kohonen clustering network for image segmentation. In *Int. Joint Conf. on Neural Networks*, volume 4, pages 2664–2667, July 1999.

[88] C. Cortes and J.A. Hertz. A network system for image segmentation. In *Int. Joint Conf. on Neural Networks IJCNN*, volume 1, pages 121–125, June 1989.

[89] S. Basu, M. Naphade, and J.R. Smith. A statistical modeling approach to content based retrieval. In *IEE Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 4, pages 4080–4083, May 2002.

[90] N. Nikolaidis and I. Pitas. Multichannel l-filters based on reduced ordering. *IEEE Trans. Circuits and Systems for Video Technology*, 6(5):470–482, April 1996.

[91] A.C. Bovik, T.S. Huang, and D.C. Munson, Jr. A generalization of median filtering using linear combinations of order statistics. *IEEE Trans. Acoustics, Speech, and Signal Processing*, 31(6):1342–1350, December 1983.

[92] M. Gabbouj, J. Astola, and S. Peltonen. Nonlinear order statistic filter design: Methodologies and challenges. In *Int. Symposium on Image Processing and Analysis*, volume 102-106, June 2001.

[93] H.M. Lin and Jr. Willson, A.N. Adaptive-length median filters for image processing. In *IEEE Int. Symposium on Circuits and Systems*, volume 3, pages 2557–2560, June 1988.

[94] E.R. Davies. On the noise suppression and image enhancement characteristics of the median, truncated median and mode filters. *IEEE Patt. Rec. Letters*, 7(2):87–97, 1988.

[95] G. Ramponi. Image processing with rational operators: noise smoothing and anisotropic diffusion. In *Int. Symposium on Image and Signal Processing and Analysis*, pages 96–101, June 2001.

[96] J. Astola, P. Haavisto, and Y. Neuvo. Vector median filters. *PIEEE*, 78:678–689, 1990.

[97] E.R. Davies. A remanent noise problem with the median filter. In *Int. Conf. on Pattern Recognition*, volume 3(C), pages 505–508, August 1992.

[98] E.R. Davies. Image distortions produced by mean, median and mode filters. *IEE Proceedings Vision, Image and Signal Processing*, 146(5):279–285, October 1999.

[99] J. Angulo and J. Serra. Morphological coding of color images by vector connected filters. In *7th Int. Symposium on Signal Processing and Its Applications*, volume 1, pages 69–72, July 2003.

[100] M. Pappas and I. Pitas. Multichannel distance filter. *IEEE Trans. Signal Processing*, 47(12):3412–3416, December 1999.

[101] J.A. Bangham, P. Ling, and R. Young. Multiscale recursive medians, scale-space, and transforms with applications to image-processing. *IEEE Trans. Image Processing*, 5(6):1043–1048, June 1996.

[102] P.E. Trahanias, D. Karakos, and A.N. Venetsanopoulos. Directional processing of color images: Theory and experimental results. *IEEE Trans. Image Processing*, 5(6):868–880, June 1996.

[103] D.G. Karakos and P.E. Trahanias. Combining vector median and vector directional filters: the directional-distance filters. In *IEEE Int. Conf. on Image Processing*, pages 171–174, October 1995.

[104] R. Lukac. Another generalisation of vector filters. In *4th EURASIP-IEEE Region 8 Int. Symposium on VIPromCom*, pages 273–278, June 2002.

[105] M. Barni. A fast algorithm for 1-norm vector median filtering. *IEEE Trans. Image Processing*, 6(10):1452–1455, October 1997.

[106] V. Chatzis and I. Pitas. Fuzzy scalar and vector median filters based on fuzzy distances. *IEEE Trans. Image Processing*, 8(5):731–734, May 1999.

[107] A. Kundu and J. Zhou. Combination median filter. *IEEE Trans. Image Processing*, 1(3):422–429, July 1992.

[108] H. Hwang and R.A. Haddad. Adaptive median filters: new algorithms and results. *IEEE Trans. Image Processing*, 4(4):499–502, April 1995.

[109] R.M. Haralick, S.R. Sternberg, and X. Zhuang. Image analysis using mathematical morphology. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 9:532–550, 1987.

[110] E.J. Breen and R. Jones. Attribute openings, thinnings, and granulometries. *Computer Vision and Image Understanding*, 64(3):377–389, November 1996.

[111] M.C. D'Ornellas, R. Van den Boomgaard, and J.-M. Geusebroek. Morphological algorithms for color images based on a generic-programming approach. In *Int. Symposium on Computer Graphics, Image Processing, and Vision*, pages 435–442, oct 1998.

[112] W.W. Boles, M. Kanefsky, and M. Simaan. Recursive two-dimensional median filtering algorithms for fast image root extraction. *IEEE Trans. Circuits and Systems*, 35(10):1323–1326, October 1988.

[113] K.R. Park and C.N. Lee. Scale-space using mathematical morphology. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(11):1121–1126, November 1996.

[114] A. Morales, R. Acharya, and Sung-Jea Ko;. Morphological pyramids with alternating sequential filters. *IEEE Trans. Image Processing*, 4(7):965–977, July 1995.

[115] J. Bosworth and S.T. Acton. The morphological lomo filter for multiscale image processing. In *IEEE Int. Conf. on Image Processing*, volume 4, pages 157–161, 1999.

[116] F.Y. Shih and P. Puttagunta. Recursive soft morphological filters. *IEEE Trans. Image Processing*, 4(7):1027–1032, July 1995.

[117] P. Kuosmanen, L. Koskinen, and J. Astola. Detail preserving morphological filtering. In *IEEE Int. Conf. on Pattern Recognition*, volume 3(C), pages 236–239, August 1992.

[118] P. Soille and H. Talbot. Directional morphological filtering. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(11):1313–1329, Nov 2001.

[119] A. Meijster and M.H.F. Wilkinson. A comparison of algorithms for connected set openings and closings. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(4):484–494, April 2002.

[120] J. Weickert, S. Ishikawa, and A. Imiya. Linear scale-space has first been proposed in Japan. *Journal of Mathematical Imaging and Vision*, 10:237–252, 1999.

[121] J.A. Bangham, P.D. Ling, and R. Harvey. Scale-space from nonlinear filters. In *IEEE Int. Conf. on Computer Vision*, pages 163–168, 1995.

[122] P.T. Jackway. Morphological scale-space. In *IEEE Int. Conf. on Pattern Recognition*, volume 3(C), pages 252–255, August 1992.

[123] L.M.J. Florack. Scale-space theories for scalar and vector images. In *Scale Space and Morphology in Computer Vision*, volume LNCS 2106, pages 193–204, 2001.

[124] M.H.F. Wilkinson. Attribute-space connectivity and connected filters. *Image and Vision Computing*, pages 1–10, July 2006.

[125] P. Salembier, A. Oliveras, and L. Garrido. Antiextensive connected operators for image and sequence processing. *IEEE Trans. Image Processing*, 7(4):555–570, April 1998.

[126] C. Lantuejoul and J. Serra. M-filters. In *IEE Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 7, pages 2063–2066, May 1982.

[127] S.T. Acton. A pyramidal algorithm for area morphology. In *IEEE Int. Conf. on Image Processing*, volume 2, pages 954–957, September 2000.

[128] N. Young and A.N. Evans. Psychovisually tuned attribute operators for pre-processing digital video. *IEE Proceedings Vision, Image and Signal Processing*, 150(4):277–286, October 2003.

[129] J. Andrew Bangham, J. R. Hidalgo, Richard Harvey, and Gavin C. Cawley. The segmentation of images via scale-space trees. In *Proc. British Machine Vision Conf.*, September 1998.

[130] M.H.F. Wilkinson and J.B.T.M. Roerdink. Fast morphological attribute operations using Tarjan's union-find algorithm. In L. Vincent J. Goutsias and D. S. Bloomberg, editors, *Mathematical Morphology and its Applications to Image and Signal Processing*, pages 311–320. Kluwer, 2000.

[131] L.J. Sartor and A.R. Weeks. Morphological operations on colour images. *Journal of Electronic Imaging*, 10(2):548559, April 2001.

[132] P. Salembier, J. Serra, and J.A. Bangham. Edge versus contrast estimation of morphological filters. In *IEE Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 5, pages 45–48, April 1993.

[133] K.R. Weber and S.T. Acton. On connected filters in color image processing. *Journal of Electronic Imaging*, 13(3):619–629, July 2004.

[134] E. Zaharescu, M. Zamfir, and C. Vertan. Color morphology-like operators based on color geometric shape characteristics. In *Int. Symposium on Signals, Circuits and Systems*, volume 1, pages 145–148, July 2003.

[135] C. Vertan, M. Zanfir, E. Zaharescu, V. Buzuloiu, and C. Fernandez-Maloigne. Non-linear color image filtering by color to planar shape mapping. In *IEEE Int. Conf. on Image Processing*, volume 1, pages 885–888, 2003.

[136] J. Chanussot and P. Lambert. Bit mixing paradigm for multivalued morphological filters. In *Int. Conf. on Image Processing and Its Applications*, volume 2, pages 804–808, July 1997.

[137] M. Wheeler and M.A. Zmuda. Processing color and complex data using mathematical morphology. In *IEE Proc. National Aerospace and Electronics Conf.*, pages 618–624, October 2000.

[138] N. Young and A.N. Evans. Psychovisually tuned area-morphology tools for improved image compression. In *Int. Symposium On Mathematical Morphology VI*, pages 185–195, 2002.

[139] E.R. Urbach, J.B.T.M. Roerdink, and M.H.F. Wilkinson. Connected rotation-invariant size-shape granulometries. In *Int. Conf. on Pattern Recognition*, volume 1, pages 688–691, August 2004.

[140] I. Vanhamel, I. Pratikakis, and H. Sahli. Multiscale gradient watersheds of color images. *IEEE Trans. Image Processing*, 12(6):617–626, June 2003.

[141] H. Talbot, C. Evans, and R. Jones. Complete ordering and multivariate mathematical morphology. In H. Heijmans and J. Roerdink, editors, *Mathematical Morphology and Its Applications to Image and Signal Processing*, pages 27–34, Amsterdam, 1998. Kluwer Academic Press.

[142] S. Gibson, R. Harvey, and G.D. Finlayson. Convex colour sieves. In *Proc. 4th Int. Conf. on Scale Space Methods in Computer Vision*, volume LNCS-2695, pages 550–563, 2003.

[143] A.N. Evans. Vector area morphology for motion field smoothing and interpretation. *IEE Proceedings Vision, Image and Signal Processing*, 150(4):219–226, August 2003.

[144] C. Vertan, M. Malciu, T. Zaharia, and V. Buzuloiu. A clustering approach to vector mathematical morphology. In *Proc. IEEE Int. Conf. on Electronics, Circuits, and Systems*, volume 1, pages 187–190, October 1996.

[145] S. Gibson, R. Harvey, and G.D. Finlayson. Evaluating a colour morphological scale-space. In *Proc. British Machine Vision Conf.*, June 2003.

[146] QHull 2003.1. The National Science and Technology Research Center for Computation and Visualization of Geometric Structures, University of Minnesota, 2003. www.qhull.org.

[147] N. Young. *Preprocessing Digital Video using Mathematical Morphology*. Phd thesis, Department of Engineering, University of Bath, 2003.

[148] B.K. Iyer and M.D. Macleod. Color image retrieval using the datasieve. In *IEE Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 2, pages 417 – 420, March 2005.

[149] C.T. Lu, D. Chen, and Y. Kou. Algorithms for spatial outlier detection. In *IEEE Int. Conf. on Data Mining*, pages 597–600, November 2003.

[150] E.R. Urbach and M.H.F. Wilkinson. Shape-only granulometries and gray-scale shape filters. In *Int. Symposium on Mathematical Morphology*, pages 305–314, April 2002.

[151] E.R. Urbach, N.J. Boersma, and M.H.F. Wilkinson. Vector attribute filters. *Int. Symposium on Mathematical Morphology*, Mathematical Morphology: 40 years on:95–104, April 2005.

[152] M.A. Ruzon and C. Tomasi. Edge, junction, and corner detection using color distributions. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(11):1281–1295, November 2001.

[153] D. Androutsos, K.N. Plataniotis, and A.N. Venetsanopoulos. Distance measures for color image retrieval. In *IEEE Int. Conf. on Image Processing*, volume 2, pages 770–774, September 1998.

[154] S. Wesolkowski and M.E. Jernigan. Color edge detection in RGB using jointly Euclidean distance and vector angle. In *Int. Conf. on Vision Interface, Tris-Rivieres, Canada*, pages 9–16, May 1999.

[155] D.R. Martin, C.C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, colour and texture cues. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(5):530–549, May 2004. http://www.cs.berkeley.edu/projects/vision/grouping/segbench/.

[156] Y. Rubner, C. Tomasi, and L.J. Guibas. A metric for distributions with applications to image databases. In *IEEE Int. Conf. on Computer Vision*, pages 59–66, January 1998.

[157] Bryan S. Morse. Brigham Young University, 1998-2000. http://homepages.inf.ed.ac.uk/-rbf/CVonline/LOCAL_COPIES/MORSE/iu.pdf.

[158] USC-SIPI Database. Southern California University, July 2006. http://sipi.usc.edu/database/.

[159] Adrian F. Clark. The PCCV project: Benchmarking vision systems. University of Exeter, 2005. http://peipa.essex.ac.uk/benchmark/databases/.

[160] Common datasets and queries in MPEG-7 color core experiments. MPEG-7 Standards Body, October 1999. Tech. Rep. ISO/IEC JTC1/SC29/WG11/MPEG99/M5060.

[161] Gerald Schaefer. UCID database. School of Computing and Informatics, Nottingham Trent University, 2005. http://vision.doc.ntu.ac.uk/datasets/UCID/ucid.html.

[162] R.J. O'Callaghan and D.R. Bull. Combined morphological-spectral unsupervised image segmentation. *IEEE Trans. Image Processing*, 14(1):49–62, January 2005.

[163] H. Zhang, J.E. Fritts, and S.A. Goldman. A co-evaluation framework for improving segmentation evaluation. In *SPIE Defense and Security Symposium - Signal Processing, Sensor Fusion, and Target Recognition XIV*, pages 420–430, March 2005.

[164] J.S. Cardoso and L. Corte-Real. Toward a generic evaluation of image segmentation. *IEEE Trans. Image Processing*, 14(11):1773–1782, November 2005.

[165] H. Lu, J.C. Woods, and M. Ghanbari. Image segmentation by binary partition tree. *Electronics Letters*, 42(17):966–967, August 2006.

[166] A.N. Evans. Morphological gradient operators for colour images. In *IEEE Int. Conf. on Image Processing*, volume 5, pages 3089–3092, October 2004.

[167] Robert Fisher. Caviar: Context Aware Vision using Image-based Active Recognition. University of Edinburgh, September 2005. http://homepages.inf.ed.ac.uk/rbf/CAVIAR.

[168] Referencing Electronic Sources, December 1997. http://www.lisa.sbu.ac.uk/helpsheets/hs31.pdf.